Introducing Blender Version 3.0.0

The Complete Guide to Blender Graphics

Computer Modeling & Animation

SEVENTH EDITION

JOHN M. BLAIN

CRC Press
Taylor & Francis Group
AN A K PETERS BOOK

The Complete Guide to Blender Graphics Computer Modeling & Animation

SEVENTH EDITION

The Complete Guide to Blender Graphics Computer Modeling & Animation

SEVENTH EDITION

JOHN M. BLAIN

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

AN A K PETERS BOOK

Seventh Edition published 2022
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742
and by CRC Press
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN
CRC Press is an imprint of Taylor & Francis Group, LLC
© 2022 John M. Blain

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

A catalog record has been requested for this book.

ISBN: 978-1-032-12841-2 (hbk)

ISBN: 978-1-032-12167-3 (pbk)

ISBN: 978-1-003-22642-0 (ebk)

DOI: 10.1201/9781003226420

Publisher's note: This book has been prepared from camera-ready copy provided by the authors.

Contents

Introduction

Download and Installation

The Author

Preamble

CH01 Understanding the Interface

1.1 Interacting with the Interface

1.2 First Interaction

1.3 Second Interaction

1.4 Getting Help

1.5 Examine the Interface

1.6 Rotating in the 3D Space

1.7 Other Objects

1.8 Using the Outliner Editor

1.9 Working in the 3D Viewport Editor (Object Mode)

1.10 3D Viewport Editor Modes

1.11 Working in the 3D Viewport Editor (Edit Mode)

1.12 Coloring Using the Properties Editor

1.13 Fire - Quick Method Example

1.14 Animation

1.15 Summary

CH02 Editors - Workspaces - Themes

2.1 Editor Types

2.2 Resizing Editors

2.3 Splitting Editors

2.4 Cancel an Editor

2.5 3D Viewport Features
2.6 Scene Manipulation
2.7 Scene Manipulation Widget
2.8 Multiple Scenes
2.9 Headers Menus and Panels
2.10 Headers and Panels
2.11 The Blender Interface Header
2.12 The Blender Screen Header
2.13 The 3D Viewport Header
2.14 The Widget Panel
2.15 The Tool Panel and Panel
2.16 Properties Editor Tabs
2.17 The Preferences Editor
2.18 3D Viewport Editor - Background Color
2.19 Workspaces
2.20 Creating New Workspaces
2.21 Themes
2.22 Saving a Theme

CH03 Navigate and Save

3.1 Files and Folders
3.2 Saving a File
3.3 Windows File Explorer
3.4 Windows File Explorer Diagram
3.5 Blender File Browse.
3.6 Opening Files
3.7 File Browser Header Features
3.8 Make a New Folder
3.9 Display Options

3.10 Saving Your Work
3.11 The Concept of Files
3.12 The Append or Link Command
3.13 Importing Objects
3.14 Activating Import File Types
3.15 Packing Data
CH04 Objects in the 3D View Editor
4.1 Modeling Workflow Philosophy
4.2 Starting a New File
4.3 Modifying the Scene
4.4 Object Mode and Edit Mode
4.5 3D Viewport Cursor
4.6 Selecting – Deselecting Objects
4.7 Adding Objects
4.8 Object Primitives
4.9 Locating the 3D Viewport Editor Cursor
4.10 Deleting Objects
4.11 Duplicating Objects
4.12 Object Mode Manipulation
4.13 The Last Operator Panel
4.14 Tool Panel – Widgets
4.15 Manipulation Units
4.16 Measuring Ruler / Protractor
4.17 Precision Manipulation
4.18 Coloring Objects
4.19 Other Types of Objects
4.20 Naming Objects
4.21 The Header Button Menus

4.22 Meta Shapes

CH05 Editing Objects

5.1 The Mesh Object

5.2 Edit Mode Selecting

5.3 Selecting Vertices, Edges and Faces

5.4 Manipulating Selected Vertices, Edges and Faces

5.5 Creating Vertices by Subdivision

5.6 Adding and Deleting Vertices, Edges, or Faces

5.7 Center Points

5.8 Joining and Separating

5.9 Creating Vertex Groups

5.10 Proportional Vertex Editing

5.11 Inset Faces

5.12 Parenting

CH06 Editing Tools

6.1 The Edit Mode Tool Panel

6.2 The Add Menu

6.3 Last Operator Panels

6.4 Extrusion

6.5 The Extrude Region Tool

6.6 Inset Faces

6.7 The Inset Faces Tool

6.8 The Bevel Tool

6.9 Edge and Loop Selection

6.10 Loop Cut Tool

6.11 The Knife Tool

6.12 The Poly Build Tool

6.13 The Spin Tool

6.14 Creating a Spin Profile
6.15 Spin Duplication
6.16 The Screw Tool
6.17 The Smooth and Randomize Tools
6.18 The Edge Slide And Vertex Slide Tool
6.19 The Shrink Fatten Tool
6.20 The Shear Tool
6.21 The Rip Region Tool

CH07 Modifiers

7.1 Modifiers in General
7.2 The Modifier Stack
7.3 The Modify Group
7.4 The Simulate Group
7.5 Generate and Deform Modifiers

CH08 Editing with Generate Modifiers

8.1 Modifiers – Generate
8.2 The Bevel Modifier
8.3 Array Modifier
8.4 Boolean Modifier
8.5 Build Modifier
8.6 Decimate Modifier
8.7 Edge Split Modifier
8.8 Mask Modifier
8.9 Mirror Modifier
8.10 Multiresolution Modifier
8.11 Remesh Modifier
8.12 Screw Modifier
8.13 Skin Modifier

8.14 Solidify Modifier
8.15 Subdivision Surface Modifier
8.16 Triangulation and Weld Modifier
8.17 Wireframe Modifier
CH09 Editing with Deform Modifiers
9.1 Modifiers for - Deform
9.2 Armature Modifier
9.3 Cast Modifier
9.4 Corrective Smooth Modifier
9.5 Curve Modifier
9.6 Displace Modifier
9.7 Hook Modifier
9.8 Laplacian Deform Modifie,
9.9 Lattice Modifier
9.10 Mesh Deform Modifier
9.11 Shrinkwrap Modifier
9.13 Simple Deform Modifier
9.14 Smooth Modifier
9.15 Smooth Corrective Modifier
9.16 Smooth Laplacian Modifier
9.17 Warp Modifier
9.18 Wave Modifier
CH10 Editing Using Curves
10.1 Curves Circles and Paths
10.2 Bezier Curve
10.3 Bezier Circle
10.4 Nurbs Path
10.5 Nurbs Circle

10.6 Modeling from a Curve
10.7 Closed Loops
10.8 Using Nurbs Curves
10.9 Nurbs Circle
10.10 Nurbs Curve
10.11 Lofting
CH11 Editing Techniques - Examples
11.1 Creating a Landscape
11.2 Dupliverts
11.3 Modeling an Aircraft
11.4 Sculpting – Sculpt Mode
11.5 Sculpting Demonstration
11.6 Creating a Humanoid Figure
CH12 The Outliner and Collections
12.1 Collections in the Outliner Editor
12.2 View Options in the Outliner Editor
CH13 3D Text
13.1 Creating 3D Text in Blender
13.2 The Object Data Properties
13.3 Fonts
13.4 Creating Text on a Curve
13.5 Converting Text to a Mesh Object
13.6 Converting Text to a Curve
13.7 Entering External Font
CH14 Viewport Shading
14.1 Viewport Shading Options
14.2 Wireframe Viewport Shading
14.3 Solid Viewport Shading

14.4 Color Display Options
14.5 Background Displays
14.6 More Solid Viewport Shading
14.7 Rendered Viewport Shading
14.8 Material Preview Shading
14.9 World Settings
14.10 Simplified Viewport Shading
CH15 Scene Lighting and Cameras
15.1 Scene Lighting
15.2 Light Type
15.3 Cameras
15.4 Camera Settings
15.5 Camera Switching
15.6 Camera Tracking
15.7 Basic Lighting Arrangement
15.8 Background Scene Lighting
15.9 Images as a Background
15.10 Volumetric Lighting
CH16 Materials - Textures - Nodes
16.1 Materiai Definition
16.2 Materials in Practise
16.3 Default Cube Material Properties
16.4 Adding Material Properties
16.5 Alternative Material Assignment
16.6 Multiple Material Datablocks
16.7 The Material Slot
16.8 The Color Picker
16.9 Material Display

16.10 Lightiing Affects
16.11 Materials using Nodes
16.12 What Are Nodes
16.13 Accessing and Viewing Node Effects
16.14 Noodle Curving
16.15 The Shader Editor
16.16 The Shader Node Menu
16.17 Adding Nodes
16.18 The Shading Workspace
16.19 Scene Arrangements
16.20 Mixing Material Example
16.21 The Principled BSDF Node
16.22 The Color Ramp Node
16.23 Saving Screen Space
16.24 Vertex Paint

CH 17 Textures

17.1 Procedural Texture
17.2 Image as a Texture
17.3 Texture Mapping
17.4 Unwrapping Options
17.5 PBR Textures
17.6 Texture Properties:
17.7 Texture Painting
17.8 Texture Displacement

CH18 Node System and Usage

18.1 The Texture Node Editor
18.2 The Geometry Node Editor
18.3 Transparency

18.4 Volumetric Lightinc
18.5 Compositing Nodes
CH19 Rendering
19.1 Rendering
19.2 Outpub Properties
19.3 The Dimensions Tab
19.4 The Output Tab
19.5 Rendering a JPEG Image
19.6 Rendering a Movie File
19.7 Video Playback
19.8 Video Codecs
19.9 Making a Movie
CH20 Animation
20.1 Animation in the 3D Viewport
20.2 Movement in the 3D View Editor
20.3 Planning the Animation
20.4 Keyframes Time and Interpolatior
20.5 Animation Speed and Length
20.6 Repositioning the Timeline Cursor
20.7 Inserting Keyframes
20.8 Scrubbing the Animatioi
20.9 Playing the Animation
20.10 Adding Keyframes
20.11 Automatic Keyframing
20.12 Controlling the Animation
20.13 The Dope Sheet Editor
20.14 The Graph Editor
20.15 Editor and Commands

20.16 Selecting the Curve GE
20.17 Rotation – Euler. Rotation
20.18 Graph Editor Cursor
20.19 Editing the Curve GE
20.20 Editing the Curve Dope Sheet
20.21 Scaling in the Dope Sheet
20.22 Other Types of Curves
20.23 The Curve Properties Panel
20.24 Animating Rotation
20.25 Rotation Using F-Curves
20.26 Animating Other Features
20.27 Keying Sets
20.28 Animation Follow Path
20.29 Evaluation Time
20.30 Keyframing
20.31 Displacement Sound Animations
20.32 Sound Effect and Cast Modifier
20.33 The Animation Workspace

CH21 Constraints

21.1 Introduction to Constraints
21.2 Track to Constraint
21.3 Constraint Stack
21.5 The Transformation Constraint
21.6 Tracking Constraints
21.7 Relationship Constraints
21.8 The Action Constraint
21.9 The Shrinkwrap Constraint
21.10 Follow Path – Extrude/Bevel

CH22 Armatures & Character Rigging

22.1 Single Bone Armature

22.2 Adding Armatures

22.3 Adding Bones

22.4 The Armature Rig

22.5 Multiple Armatures

22.6 Child Parent Relationship

22.7 Armature Display Types

22.8 Building an Armature Rig

22.9 The Extrude Tool

22.10 Deforming a Mesh

22.11 Parenting with The Armature Modifier

22.12 Assigning Vertices – Vertex Groups

22.13 Assigning Vertices-Set Parent

22.14 Assigning Vertices-Weigh Paint

22.15 Inverse Kinematics Constraint

22.16 Spline IK Constraint

22.17 Forwara

22.18 Character Rigging

22.19 Bone Naming

22.20 Assigning the Mesh

22.21 Vertex Groups

22.22 Posing the Character Model

22.23 Pre-Assembled Armatures

CH23 Shape Keys & Action Editors

23.1 Shape Key Editor

23.2 Set Limits of Movement

23.3 Inserting Keyframes

23.4 Inserting Multiple Keyframes
23.5 The Animation
23.6 Additional Keys
23.7 Action Editor
23.8 Shape Keys and Action Editor in Practise
CH24 Particle Systems
24.1 The Default Particle System
24.2 The Emissions Tab
24.3 The Emission Source Tab
24.4 The Cache Tab
24.5 The Velocity Tab
24.6 Particle Display
24.7 Particle Emission Options
24.8 Order of Emission
24.9 Normals
24.10 Particle Modifiers
24.11 Particles Array
24.12 The Viewnort Display Tab
24.13 Particle Interaction
24.14 Wind Force Effect
24.15 Boids Particles
24.16 Hair Particles
24.17 Particles Edit Mode
24.18 Particles for Arrays
24.19 More Arrays
24.20 The Assignment Panel
24.21 Paticie Exercises
24.22 Multiple Particle Systems

24.23 Keyed Particles
CH25 Physics and Simulation
25.1 Applying and Cancelling Physics
25.2 Real World Physics
25.3 Force Fiele
25.4 Collision Physics
25.5 Cloth Physics
25.6 Soft Body Physics
25.7 Fluid Simulation
25.8 Quick Methods
25.9 Fluid Simulation Continued
25.10 Fluid Simulation Experiment
25.11 Smoke and Fire Simulation
25.12 Smoke and Fire from Scratch
CH26 Dynamic Paint
26.1 Dynamic Paint – Painting
CH27 Installing Add-Ons
27.1 Blender Bendy Bone Add-On
27.2 Obtaining the Add-on File
27.3 Installing the Add-On
27.4 Using the Add-On
CH28 Grease Pencil 2D Animation
28.1 Introduction
CH29 Making a Movie
29.1 Making a Movie
29.2 Storyboare
29.3 The Video Files
29.4 The Sound File

29.5 Video Editing Workspace
29.6 The File Browser Editor
29.7 Preparation
29.8 Video Sequence Editor
29.9 Rendering the Movie
29.10 Additional Features
CH30 Drivers
30.1 Blender Drivers Introduction
30.2 The Drivers Editor
30.3 Understanding Drivers
30.4 Randomize Object Properties
30.5 Using Random.Py.blend
30.6 Material Property Drivers
30.7 Duplicating the Object
CH31 Cycles & Workbench Render
31.1 Cycles Render
31.2 How to Start Cycles
31.3 Create an Object Light Source
31.4 Cycles in Practise
31.5 Workbench Render
Appendix A Basic Blender Commands
Appendix B Internet Resources
Index

Introduction

The Program and the Book

The Book — The Complete Guide to Blender Graphics - 7th Edition

The Complete Guide to Blender Graphics - 7th Edition provides instruction in the use of the Computer Graphics 3D Program **Blender.** The book is an operation manual for those who wish to undertake a learning experience and discover a wonderful creative new world of computer graphics. The book also serves as a reference for established operators.

Instructions throughout the book introduce Blender's features with examples and diagrams referenced to the **Graphical User Interface (GUI).**

The Complete Guide to Blender Graphics originated when Blender's Graphical User Interface was transformed with the release of Blender version 2.50. Subsequent editions of the book have kept pace with developments to the program and have included new material.

The **7thEdition of The Complete Guide to Blender Graphics** is applicable to **Blender Version 2.93.4 LTS \ 3.0.0** incorporating changes to the program such as a modified application of Materials (color) and Textures, the inclusion of Geometry Nodes and revised procedures for using Drivers.

For new users this book provides a fantastic learning experience in **Computer Graphics** using **Blender,** by introducing the operation of the Blender program through the use of its' Graphical user Interface. The book is intended to be read in conjunction with having the program in operation, with the interface displayed.

Instruction is presented using the tools displayed in the Graphical User Interface, with basic examples demonstrating results. Understanding where tools are located, their uses and how they are implemented will allow the

reader to more easily follow detailed instruction in the many written and video tutorials available on the Internet.

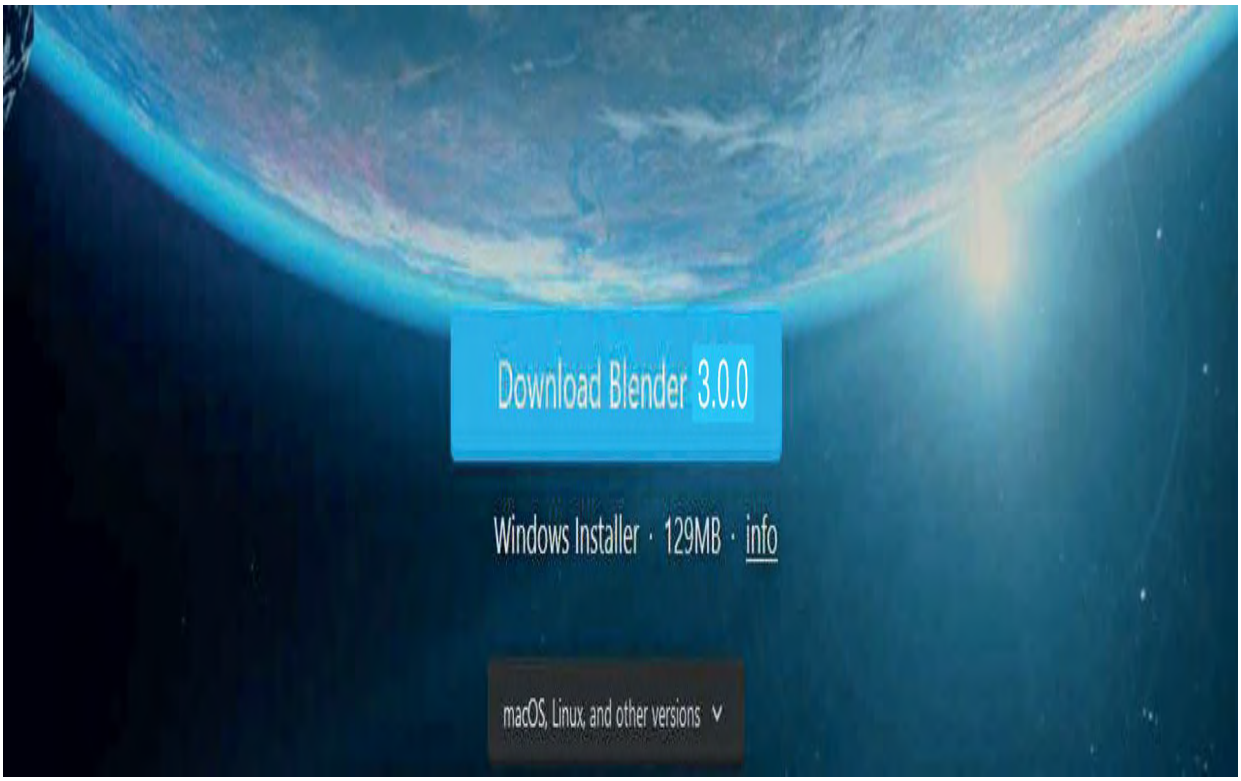
The Program - Blender

Blender is a 3D Computer Graphics Program with tools for modeling and animating objects and characters and creating background scenes. Scenes may be made into still images. Animated sequences may be used for video production. Models and Scenes are enhanced with color and texture producing brilliant realistic effects. The still images and video may be for artistic appreciation or employed as architectural or scientific presentations. There are also tools for 2D animation production. Stand alone models may be used for 3D Printing.

The Blender program is maintained by the **Blender Foundation** and released as **Open Source Software** which is available for download and **FREE** to be used for any purpose.

The program may be downloaded from:

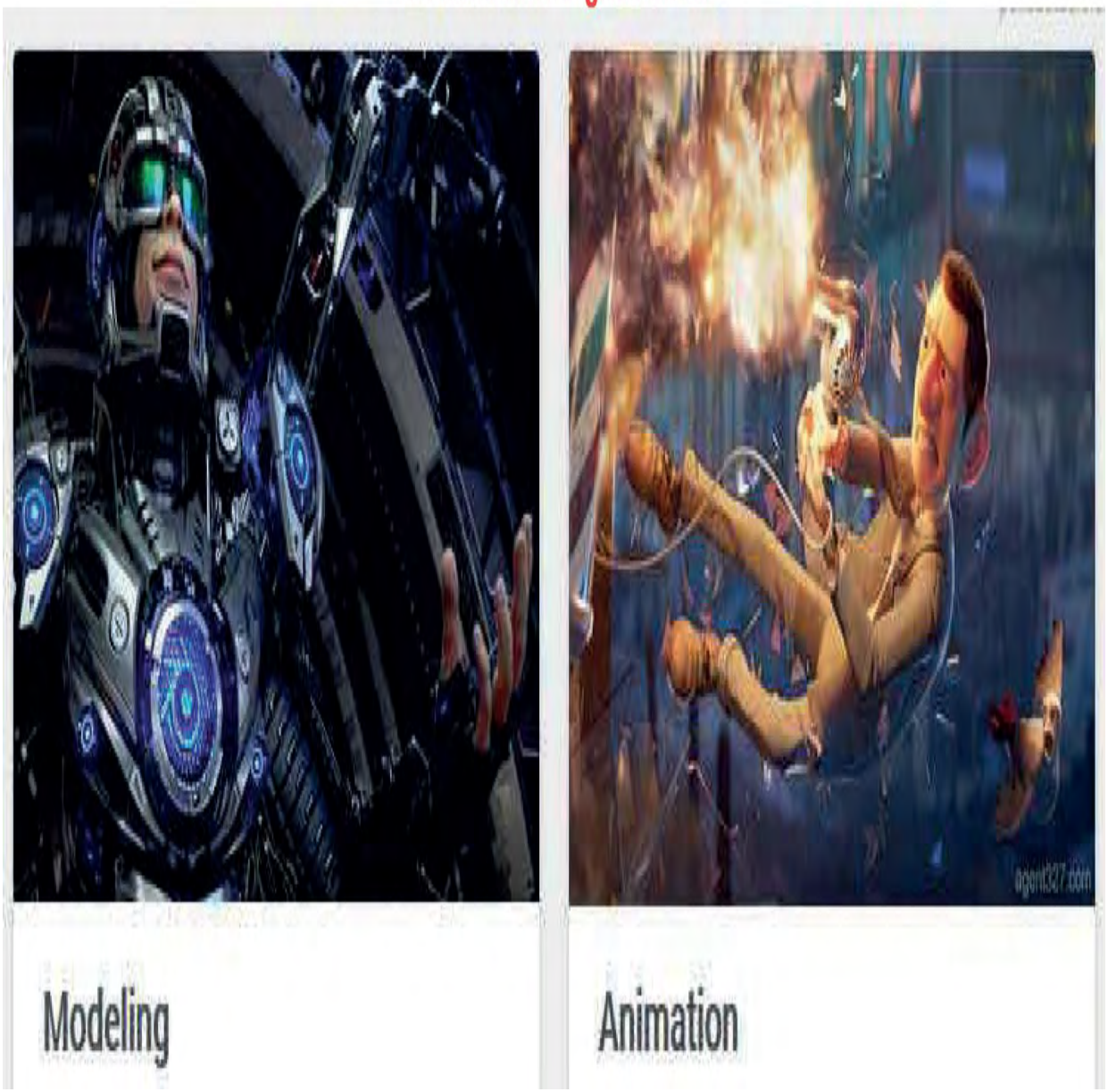
www.blender.org



Blender Features

A comprehensive display of the Blender features is available at:

www.blender.org/features/



www.blender.org/features/

Blender Platforms

A **computing platform** or **digital platform** is the environment in which a piece of software is executed. It may be the hardware or the operating system (OS)

Blender is a cross platform application for **Windows, Linux** and **MacOS** operating systems.

The operation of Blender in this manual is applicable to all operating platforms but operations ancillary to the program, such as, saving work to the computers hard drive, have been described exclusively using a Windows operating system.

System Requirements

Hardware Requirements

Minimum	Recommended	Optimal
64-bit dual core 2Ghz with SSE2 support	64-bit quad core CPU	64-bit eight core CPU
4 GB RAM	16 GB RAM	32 GB RAM
1280 x 768 display	Full HD display	Full HD display
Mouse, trackpad or pen + tablet	Three button mouse + tablet	Three button mpuse + tablet
Graphics card with 1 GB RAM, OpenGL 3.3 RAM	Graphics card with 4 GB RAM	Graphics card with + 12 GB

Supported Graphics Cards

Always make sure to install the latest drivers from the graphics card manufacturer website. These requirements are for basic Blender operation. Cycles rendering using the GPU has higher requirements.

NVIDIA

GeForce 400 and newer. Quadro Tesla GPU architecture and newer, including RTX-based cards, with NVIDIA drivers (list of all GeForce and Quadro GPUs).

AMD

GCN 1st gen and newer. Since Blender 2.9.1, Terascale 2 architecture is fully depreciated, try using 2.90 (albeit not supported)

Intel

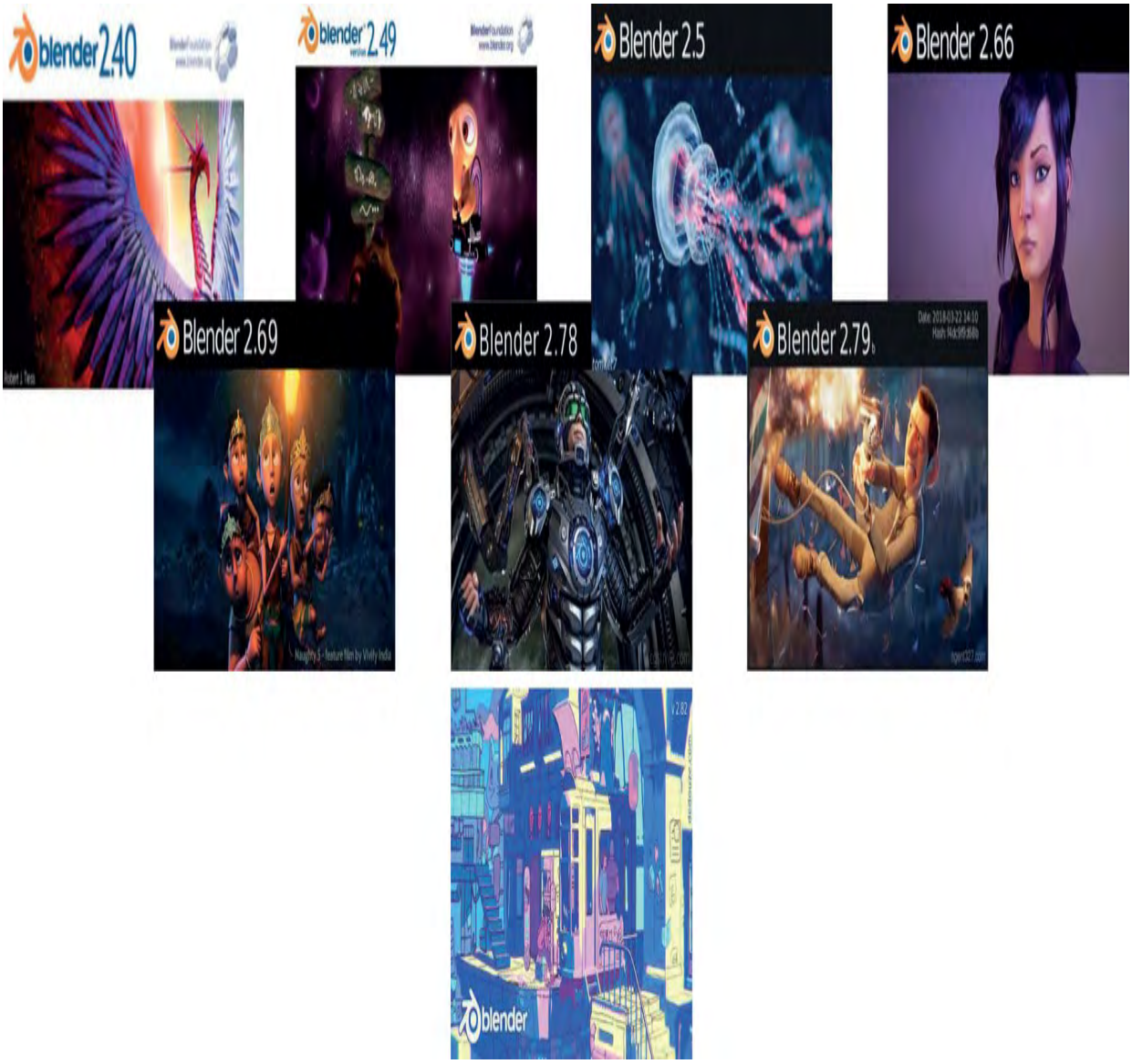
Haswell architecture and newer. [list of all Intel GPUs]

macOS

Version 10.13 or newer for Intel processors on supported hardware. Version 11.0 for Arm-based processors (Apple Silicon).

Program Evolution

Blender is continually evolving. New versions of the program are released as additions and changes are incorporated, therefore, it is advisable to check the Blender website, from time to time.

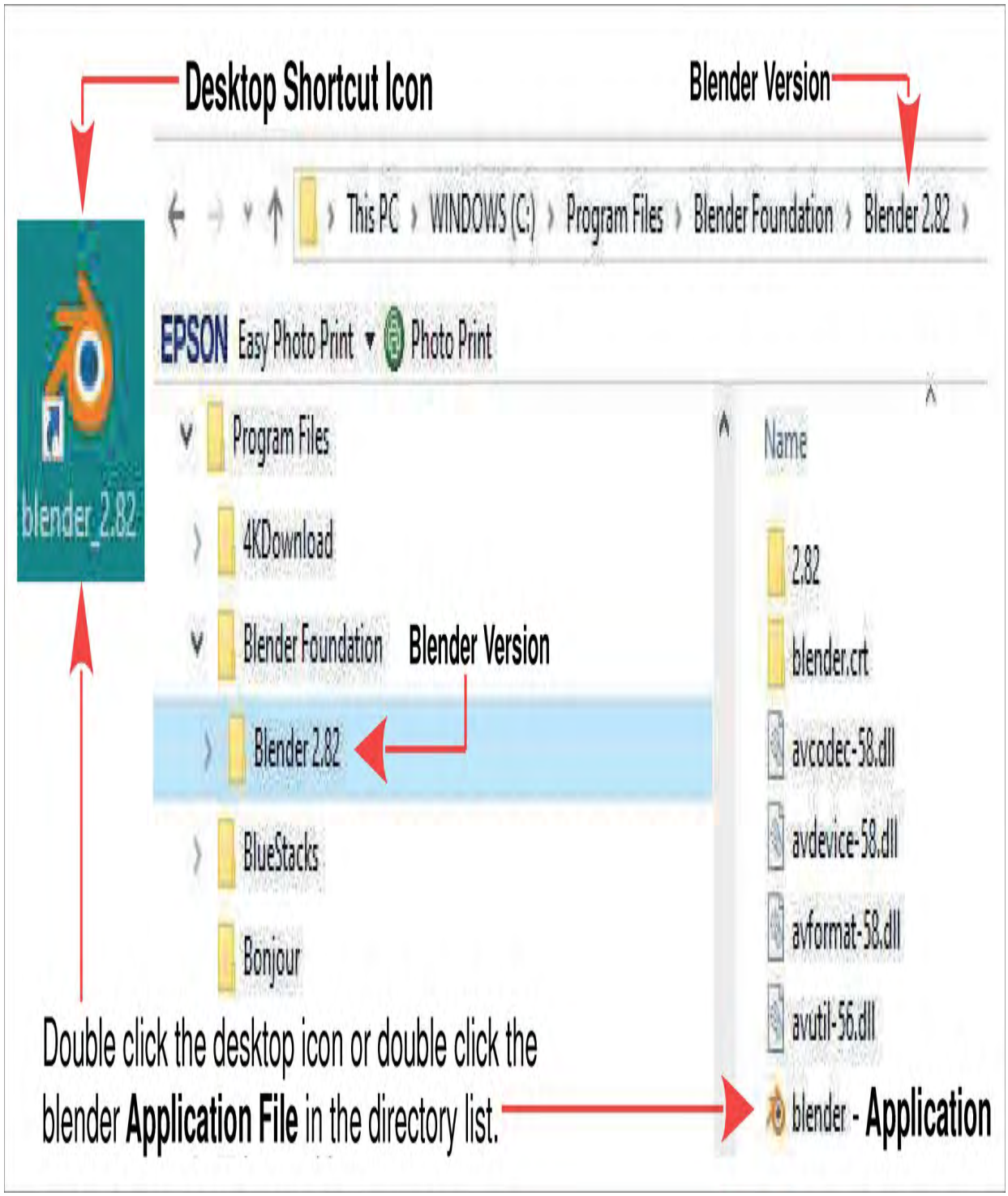


Earlier versions of the program and documentation may be obtained which provide valuable information when you are conversant with the current release of the program. Video tutorials available on the internet also provide

valuable information but may not strictly adhere to the current user interface or work flow. Major transformations occurred when the program changed from version 2.49 to 2.50 and again at the change from version 2.79 to version 2.82. Since 2.82 development has continued.

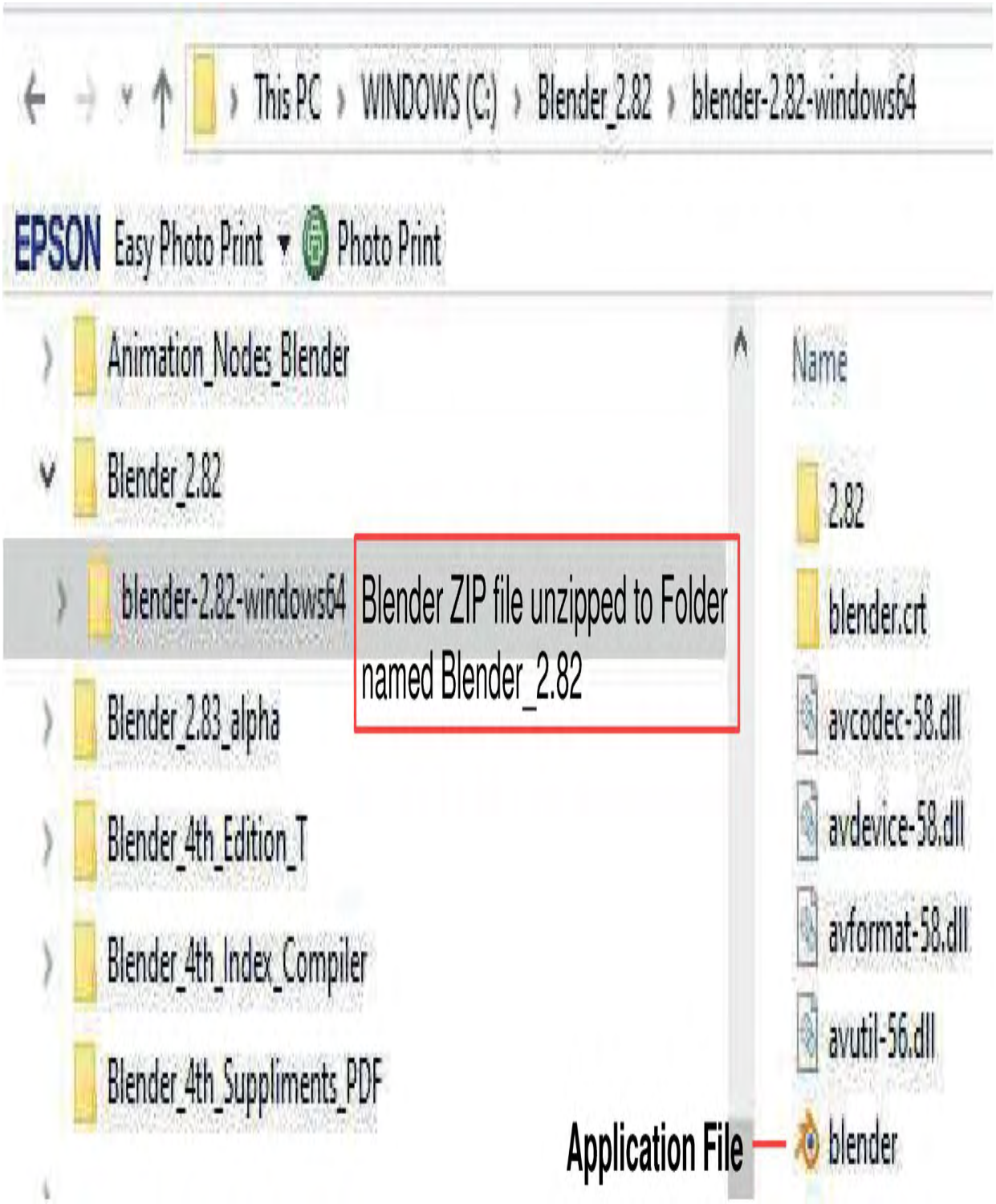
Starting the Program

How you start Blender depends on how you have installed the program (see Download & Installation following). If you have used the **MSI installer option** for Windows, Blender will be in the Program Files directory on your C: Drive and a shortcut icon will have been placed on your desktop. If you have installed to a Windows operating system, Blender will be listed under, Program Files\ Blender Foundation\ Blender.



If you have downloaded and unzipped the compressed (ZIP) file for Blender the **blender.exe** application file will be located in the folder where

you unzipped the compressed file. Open the folder and double click **blender.exe** or right click and select **Open.**



Note: By having one version of Blender installed via the Installer(.msi) option and another using the ZIP method you can have more than one Blender version installed on your computer at the same time. This is useful for version comparison or for development purposes. The diagrams depict Windows 10 File Explorer. At the time of writing Windows 10 is the current version with Windows 11 forcast for release. It is anticipated that the Windows 11 File Explorer will be similar.

Shortcut

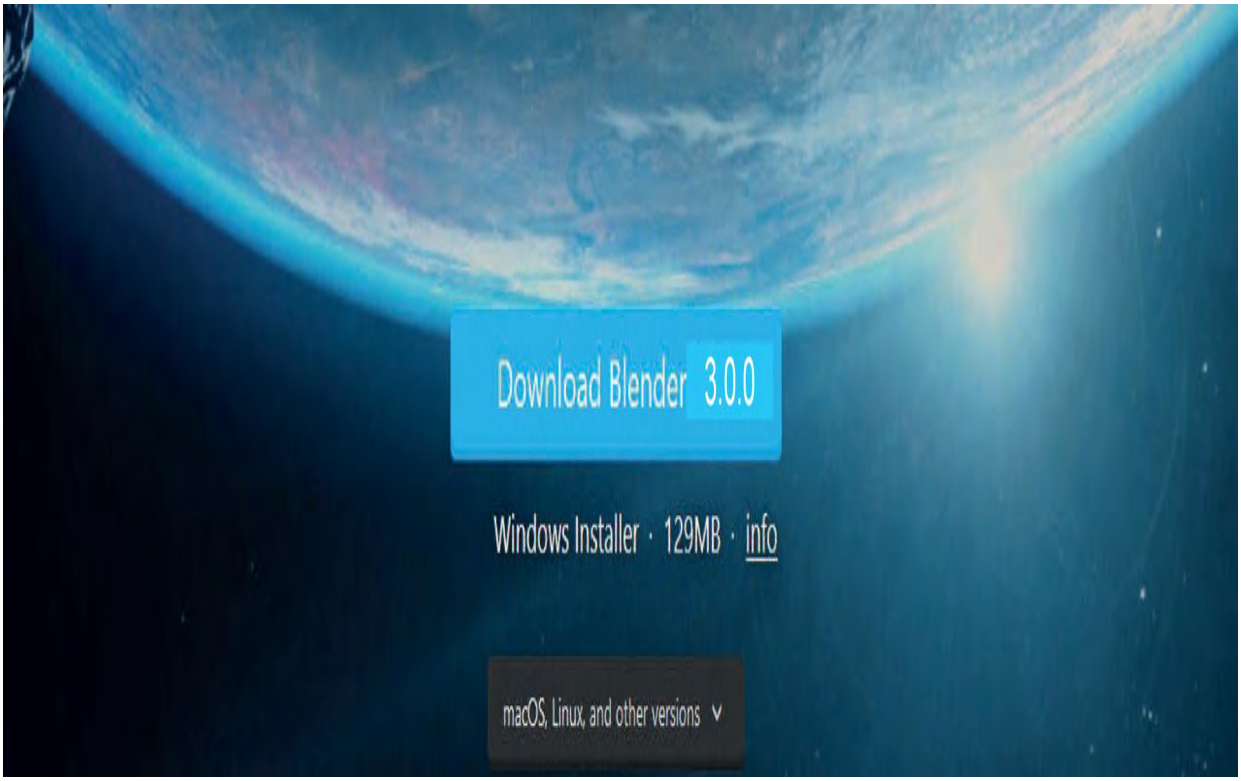
In the directory containing the **blender.exe** file create a shortcut and place it on the desktop.

The Manual Comphation

This manual has been compiled with reference to program builds leading to the release of Blender Version 3.0.0. As new builds are released, subtle changes improving the program's interface are implemented and new program features are incorporated. Every effort has been made to include these changes in diagrams and instructions which demonstrate operational features of the program.

Images used to construct diagrams may occasionally differ to what you see on your computer screen. The Blender screen display may be customized or modified to suit individual user preferences. There are several in built display themes which you may choose. In some cases the screen display has been altered to facilitate the construction of diagrams (Figures) and enhance visibility of construction lines. When alterations have been made they do not detract from the instruction presented.

Download & Installation



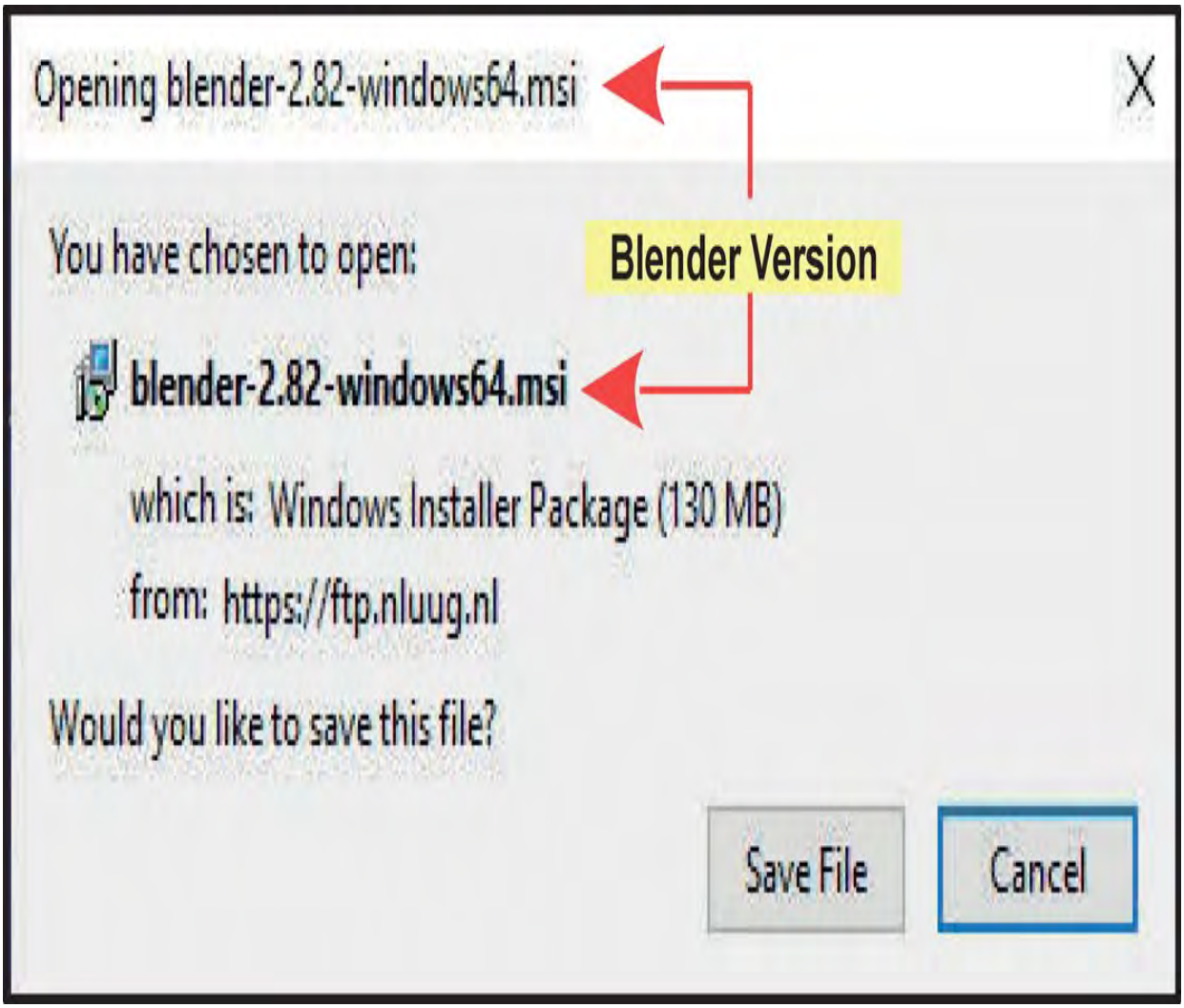
Download Blender

Download Blender from:

www.blender.org

Select the current Blender version which is applicable to your operating system. Blender is available for Windows, MacOS, Linux and Steam.

The download options are shown in the previous diagram. Selecting the Windows Version presents the MSI Installer which you save to your PC.



With a Windows operating system the MSI Installer will be saved to the Downloads Folder.

Note: The diagram shows the download file for Blender version 2.82. The current version is Blender 3.0.0.

Installation on a Windows Operating System

Installing with the Installer(.msi) Option

Double click on the file name in the Downloads folder, follow the prompts and Blender will be automatically installed to the **Program Files** folder on your computer and an icon will be placed on your **Desktop.**

Installing with the ZIP Option

In some cases you can download a compressed ZIP File instead of an MSI Installer.

With a ZIP file you have to unzip the file. You first create a new folder on your computers hard drive then use a program like 7-Zip or Win-Zip to unzip (decompress) the zip file into the new folder (see the note at the end of the chapter).

When the file is unzipped into the new folder you will see **blender.exe** as one of the entries. You double click on this to run Blender or you create a shortcut which places an icon on your desktop.

When using either installation option you double click the **blender.exe** file to run the program. Shortcuts on the Desktop are shortcuts to the blender.exe file.

Note: By having one version of Blender installed via the Installer(.msi) option and another using the ZIP method you can have more than one Blender version installed on your computer at the same time. This is useful for version comparison or for development purposes.

Installing Blender on a Linux Operating System

macOS

https://docs.blender.org/manual/en/latest/getting_started/installing/macos.html

Linux

https://docs.blender.org/manual/en/latest/getting_started/installing/linux.html

The Author

John M. Blain was born in 1942 in Swindon, Wiltshire in England and emigrated to Canada with his family in 1952. He now lives in Coffs Harbour,

New South Wales in Australia.

Drawing and painting were skills John developed from an early age and while attending school on Vancouver Island he became interested in wood sculpture inspired by the work of the indigenous west coast people. Artistic pursuits were curtailed on graduating from high school when he returned to England to undertake a technical engineering apprenticeship. Following his apprenticeship he worked for a short period in England then made the decision to return to Vancouver, Canada. On the voyage between Southampton and Vancouver he met his wife to be and Vancouver became a stopover for a journey to Sydney Australia. In this new country he began work as an engineering draughtsman, married, had children and studied engineering . The magic milestone of seven years saw John with his young family move out of the city to the coastal town of Coffs Harbour, New South Wales.

Coffs Harbour was a center for sawmill machinery and John became engaged in machinery design and manufacture. He acquired a sound knowledge of this industry acting as installation engineer then progressing to sales. This work afforded travel throughout Australia, Canada, the United States and New Zealand.

On retirement, artistic pursuits returned with additional interests in writing and computing. Writing notes whilst learning computer animation using Blender resulted in **The Complete Guide to Blender Graphics.** The first edition, published in 2012, was well received and encouraged John to compile a second edition in line with the latest version of the Blender program. This afforded the opportunity to include new material. Subsequent editions have followed until this new reformatted seventh edition.

Preamble

The objective of the Preamble is to explain terminology and introduce diagrams which demonstrate instructions.

Basic Objective – Graphics Program

The fundamental objective in using a computer graphics program, such as Blender, is to produce a display on a computer Screen which converts **(Renders)** into a digital image or series of images for an animation sequence. The display may only contain a single inanimate model such as that used for 3D printing but will will usually contain multiple 3D models of animate and or inanimate Objects. The arrangement of **Objects** constitutes a **Scene.** Animate Objects (animated Objects) are the moving characters in animation sequences. Inanimate Objects are the components of a Scene with which the characters interact. These may be obstacles in a Scene such as, ground planes, terrain and background.

Assumption

Before you begin to read this book it is assumed you know how to operate a computer. In the past this assumption meant you knew how to operate using a keyboard and mouse. Today many of you will be more familiar with touch screens or laptop touch pads, therefore, although this may appear to be a retrograde step the first instruction will be to familiarise you with Mouse and Keyboard operations.

Blender has been designed to be operated using a Keyboard and Mouse and instruction will be provided using these devices. You may of course adapt a drawing tablet and stylus.

Formats Conventions and Commands

In writing this book the following format conventions have been adopted:

Paragraphs are separated by an empty line and have not been indented.

Key words and phrases are printed in **bold text** with the first letter of a component name specific to Blender capitalised.

Headings are printed in **Bold Olive Green.**

The following conventions will be used when giving instructions.

When using a Mouse connected to a computer, the commands will be:

Click or **Click LMB —** In either case this means make a single click with the left mouse button with the Mouse Cursor positioned over a control on the computer Screen.

In some instances it is explicit that the left mouse button should be used.

A Control: Is a designated area on the computer Screen represented by an icon in the form of a button or bar, with or without text annotation.

Double Click — Make two clicks in quick succession with LMB (the left mouse button).

Click, Hold and Drag — Click the left mouse button, hold it depressed while moving the mouse. Release the button at the end of the movement.

Click RMB — Click the right mouse button.

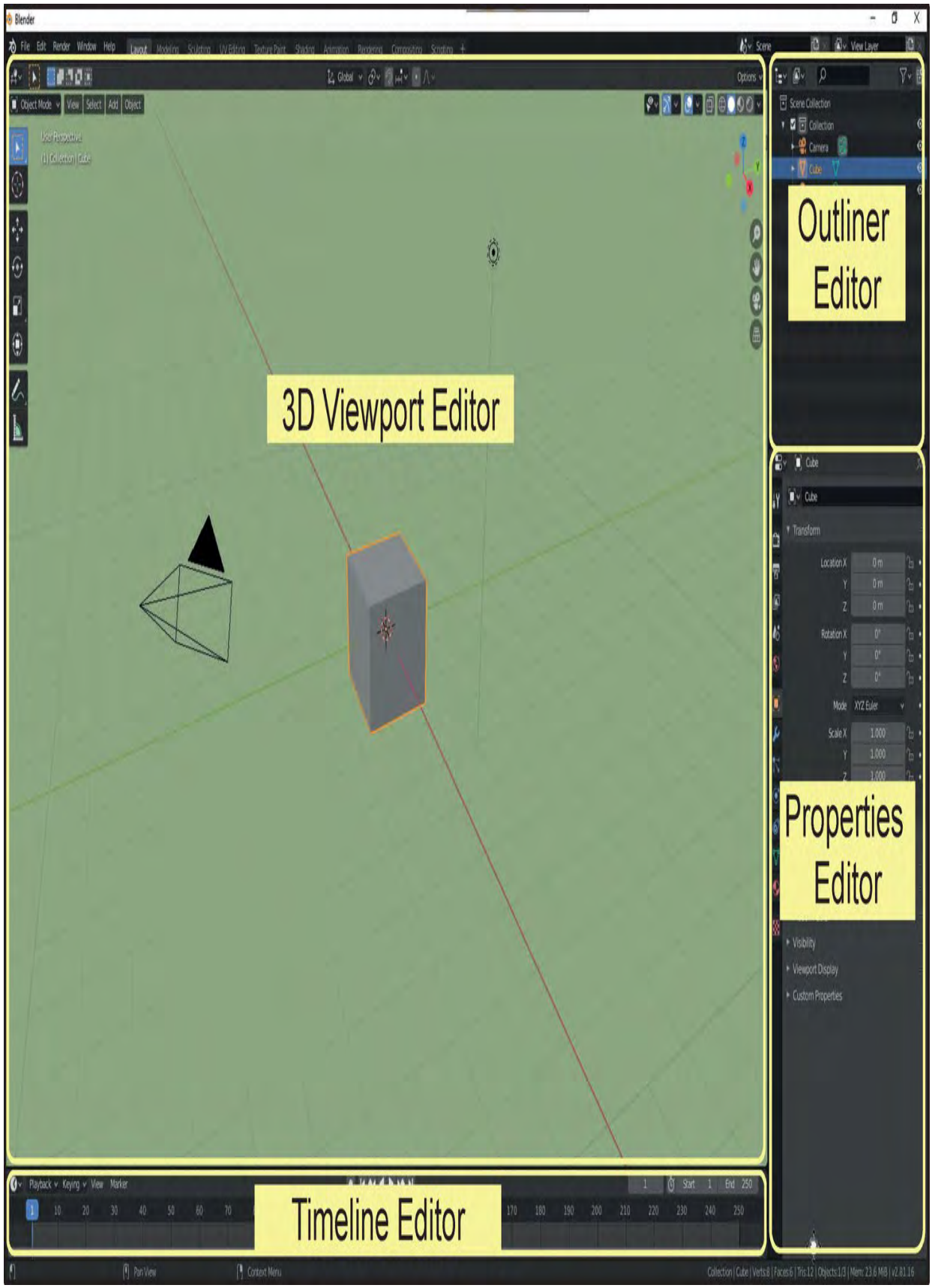
Click MMB — Click the middle mouse button (the middle mouse button may be the scroll wheel).

Scroll MMB — Scroll (rotate) the scroll wheel (MMB).

Clicking is used in conjunction with placing the Mouse Cursor over a button, icon or a slider which is displayed on the Screen.

The Grapnical User Interface (GUI)

When Blender is first opened what you see on the computer Screen is the **Graphical User Interface (GUI)** for the program. This arrangement of panels is the interface which allows you to communicate with the program by entering commands (data) using the Keyboard and Mouse, previously described. The panels that you see are called **Editors.**


3D Viewport Editor
Outliner Editor
Properties Editor
Timeline Editor

Editors

Editors (the panels in the **GUI)** are so named since the basic philosophy in operating the program is; You are presented with a set of default data producing a Screen display. You edit or modify the default data to create what you want.

There are numerous Editors for selection depending on the particular feature of Blender you wish to use. The different Editors will be introduced as features of the program are encountered.

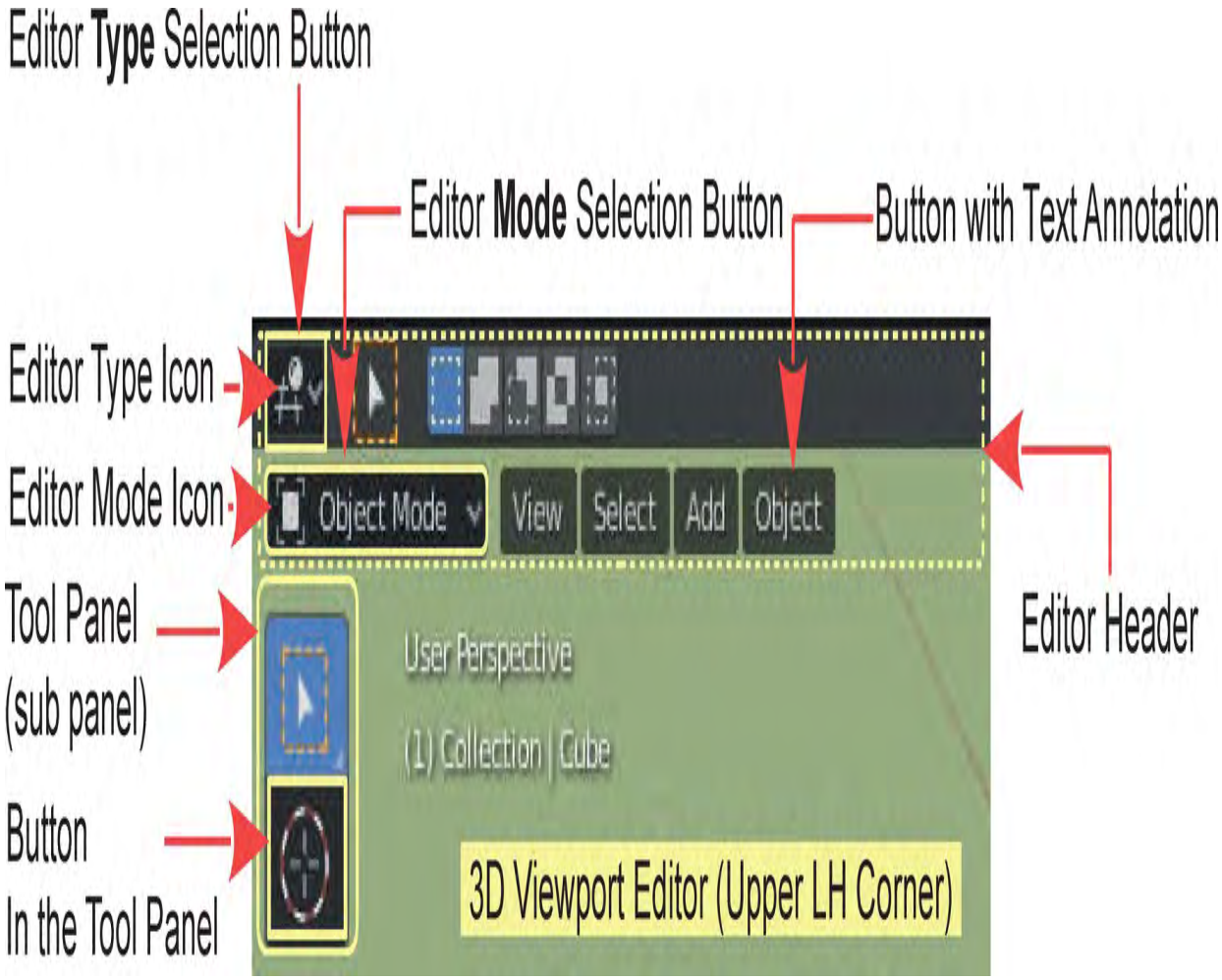
Editors interrelate since what occurs in one will affect what occurs in another. This interaction between Editors is particularly evident when considering the relationship between the 3D Viewport Editor and the Properties Editor. Editing data (entering or modifying) in the Properties Editor affects what is displayed in the 3D Viewport Editor.

Data is entered or modified by activating (clicking) controls in an Editor.

Controls - Buttons, Icons and Sliders

Each Editor in the **GUI** is a separate panel comprising a **Header** at the top of the panel and sub-panels which display within the Editor. The Header and sub-panels contain buttons which you click to activate functions or display menus for selecting functions. The buttons are displayed as text annotation, icons and panels. Each of these, relay data to the program to perform an action.

Example 1 : The 3D Viewport Editor (the default Screen display — **Upper LH Side)**

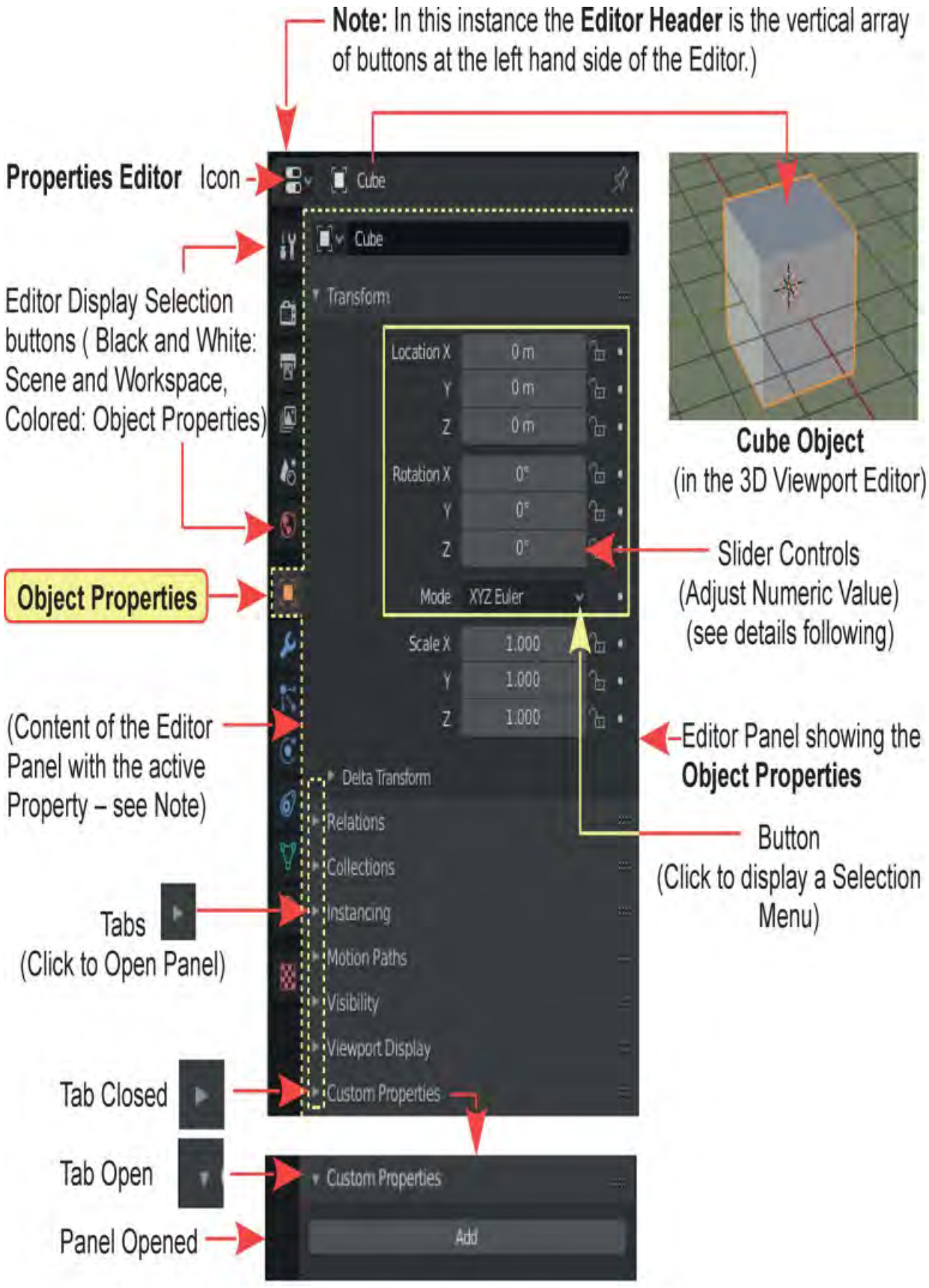


Note: The buttons shown in the diagram can be seen in the panel at the upper left hand side of the **default** Blender Screen arrangement. A detailed description of the Screen Arrangement with its Editors and panels constituting **Blenders GUI** (Graphical User Interface) is presented in **Chapter 1.**

Note: In giving instructions, **Default** means, that which is displayed on the computer Screen before any action is taken.

Example 2: The Properties Editor (the default Screen display — Lower RH Side)

The default display shows the content of the **Properties Editor** with the **Object Properties** active. In this state the controls affect the default **Cube Object** in the 3D Viewport Editor.


Note: In this instance the Editor Header is the vertical array of buttons at the left hand side of the Editor.)
Properties Editor Icon
Cube
Cube
Editor Display Selection buttons (Black and White: Scene and Workspace, Colored: Object Properties)
Transform
Location X 0 m
Y 0 m
Z 0 m
Rotation X 0°
Y 0°
Z 0°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000
Object Properties
(Content of the Editor Panel with the active Property – see Note)
Delta Transform
Relations
Collections
Instancing
Motion Paths
Visibility
Viewport Display
Custom Properties
Tabs
(Click to Open Panel)
Tab Closed
Tab Open
Panel Opened
Custom Properties
Add
Cube Object
(in the 3D Viewport Editor)
Slider Controls
(Adjust Numeric Value)
(see details following)
Editor Panel showing the Object Properties
Button
(Click to display a Selection Menu)

Note: The Editor display indicated by the broken yellow line shows controls for the active Property. In the diagram the **Object Properties** is active.

A **Button** in Blender can be a small square or rectangular area on the screen or an elongated rectangle in which case it may be referred to as a bar. Some buttons display with icons.

An **Icon** is a pictorial representation of a function. In the diagram the icon in the upper left hand corner indicates that the **Properties Editor** is displayed.

A **Slider** is an elongated area, usually containing a numeric value, which is modified by clicking, deleting and retyping the value, or clicking, holding and dragging the Mouse Cursor that displays on **Mouse Over,** left or right to decrease or increase the value. Some sliders have a small arrow at either end which display when the Mouse Cursor is **positioned over the Slider** (Mouse Over). Click on an arrow to incrementally alter the value. Some sliders directly alter the display on the computer Screen.

Slider Control Detail

Slider Controls in the Transform Tab affect the position of the Cube (the selected Object) in the 3D Viewport Editor.

With the Cube Object selected in the 3D Viewport Editor, altering the X Location Slider value to 3m and the Z Axis Rotation Slider to 19.8° moves the Cube forward along the X Axis (Red Line) and rotates the Cube about the vertical Z Axis.

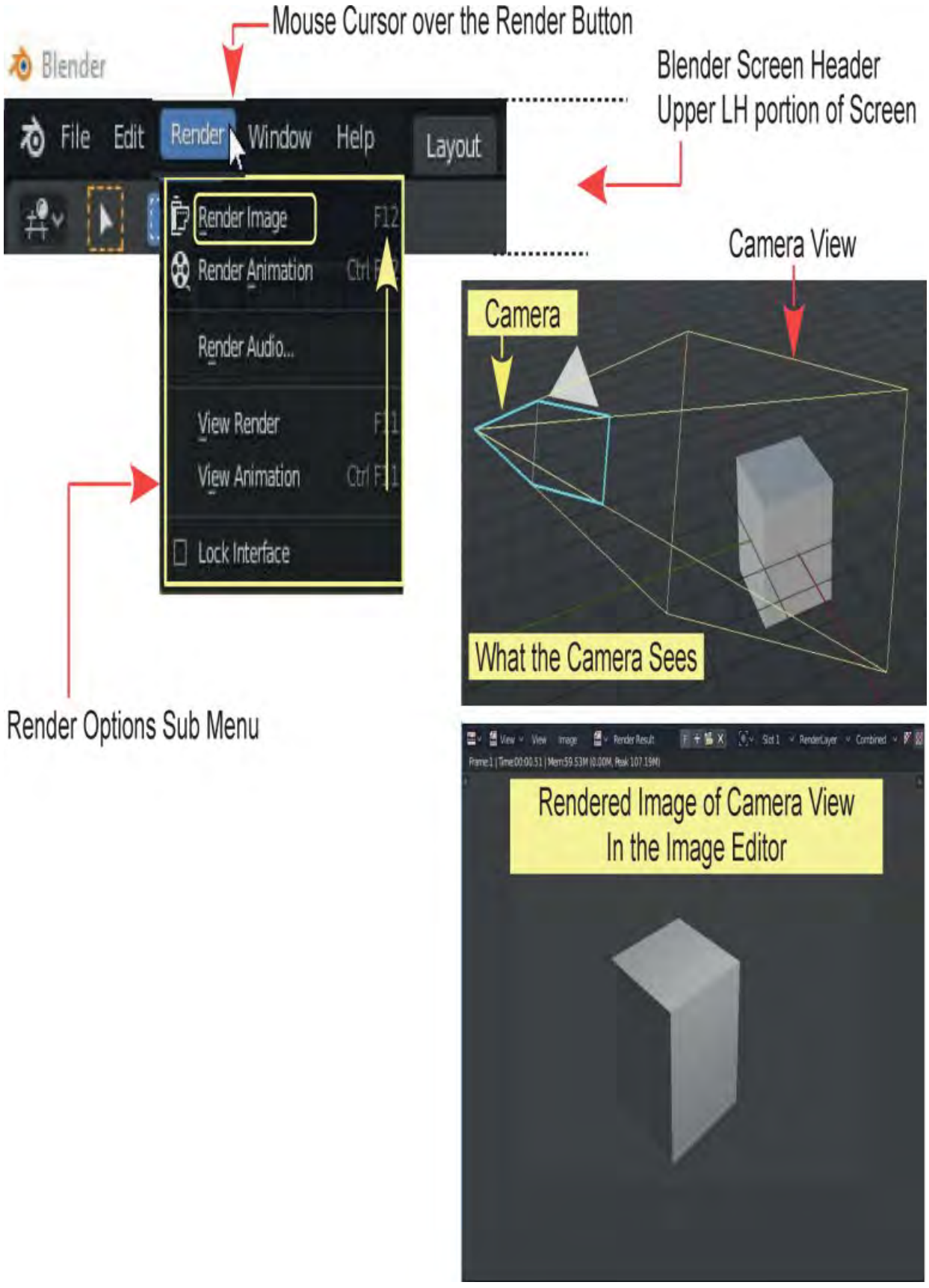
For Keyboard inpi. , a command is; to press a specific Key or a series of Keys. Press **Shift + Ctrl + T Key** means, press and hold both the **Shift** and **Ctrl** Keys simultaneously and tap the **T Key. Num Pad** (Number Pad) Keys are also used in which case the command is Press **Num Pad 0** to **9** or **Plus** and **Minus.**

Command Instruction Example:

Go to the **Blender Screen Header, Render Properties,** click **Render Image:**

Remember: A control button, icon or slider which is displayed, indicates a specific location on the computer Screen. Positioning the Mouse Cursor at this location and clicking the Mouse button or depressing a keyboard button, inputs a signal to the computer. The interpretation , made by by the computer is; signal received at specific location = perform explicit computation and export result.

The example above means, in the **Blender Screen Header,** position the **Mouse Cursor** over the **Render Properties** button and click the left mouse button, clicking once. In this case the signal received by the computer with the Mouse Cursor at the position of the Render Properties button tells the computer to display the **Render Options Sub Menu.** Positioning the Mouse Cursor over **Render Image** in the sub menu and clicking once renders an image of **Camera View** (what the camera sees). The rendered image is displayed in a new Editor panel, the **Image Editor.** The image may be saved from this location but for the time being press **Esc** on the Keyboard to cancel the render and return to the 3D Viewport Editor.


Mouse Cursor over the Render Button
Blender
Blender Screen Header
Upper LH portion of Screen
File
Edit
Render
Window
Help
Layout
Render Image
F12
Render Animation
Render Audio...
View Render
View Animation
Lock Interface
Camera View
Camera
What the Camera Sees
Render Options Sub Menu
Rendered Image of Camera View
In the Image Editor

Note: The annotation **F12** adjacent to **Render Image** is a **Keyboard Shortcut.** Pressing F12 on the Keyboard will also Render the Camera View.

Remember: The purpose of exercises in this preamble are to familiarise you with commands and diagrams not to provide instruction an any particular task. Full instruction will be provided later.

Book Work Flow

The initial work flow in the book will introduce the Editors and panels which make up the **Graphical User Interface (GUI)** and familiarise you with basic control operations. During the initial introduction detailed explanation of the Blender processes will be limited to a need to know basis. To start with, you will have to blindly follow along without understanding why. Explanation will be given as you progress and are made aware of the different Blender features.

In demonstrating one of the previous **Command Examples** the command was; **Click the Render button.**

Rendering

Rendering: The Definition from the Wiki when specifically applied to computer graphics follows. The Wiki? The Free Encyclopedia **Wikipedia.**

Https://en.wikipedia.org/wiki/Rendering_(computer_graphics)

Rendering or **Image Synthesis** is the automatic process of generating a photo realistic or non-photo realistic image from a 2D or 3D model (or models in what collectively could be called a Scene file) by means of a computer program. Also, the results of displaying such a model can be called **Rendering.** A Scene file contains objects in a strictly defined language or data structure; it would contain geometry, viewpoint, texture, lighting, and shading information as a description of the virtual Scene. The data contained in the scene file is then passed to a rendering program to be processed and output to a digital image or raster graphics image file. The term "rendering" may be by analogy with an "artist's rendering" of a Scene.

Render Engines - GUI Versions in Blende.

Render Engines are the parts of the Blender program that produce the Screen display and convert the display into an image or sequence of images. Image

sequences generate animations which in turn produce movie files.

In Blender 3.0.0 there are three **Render Engine options.** With the selection of each Render Engine type the Graphical User Interface (GUI) is displayed in a slightly different manner. Which option you chose depends on the particular process, to which the engine type is suited.

The Render Engines in Blender 3.0.0 are named; **Eevee Render, Cycles Render, Workbench Render.**

Eevee Render

The default Render Engine presented when Blender starts is **Eevee.** This is an acronym for *"Extra Easy Virtual Environment Engine".* Eevee can display a real time rendered view (depending on the Viewport Shading option selected — explanation to follow). In other words, what you see on the Screen as you make changes, is a good approximation of what you get in your final image view. Eevee quickly renders the Scene as you work but the quality of the render can incur a time disadvantage in the advanced stages of modeling.

Cycles Render

Cycles Rendering is specifically designed to produce a photo realistic high quality display of an image or frame in an animation incorporating colors, textures and special lighting. The quality of the display is adjustable since high resolution rendering comes at a cost with respect to time.

Workbench Render

Workbench Rendering uses the 3D View's drawing for quick *preview* renders. This allows you to inspect your animation (for object movements, alternate angles, etc.). This can also be used to preview your animations — in the event your Scene is too complex for your system to play back in real-time

in the 3D View. You can use Workbench to render both images and animations.

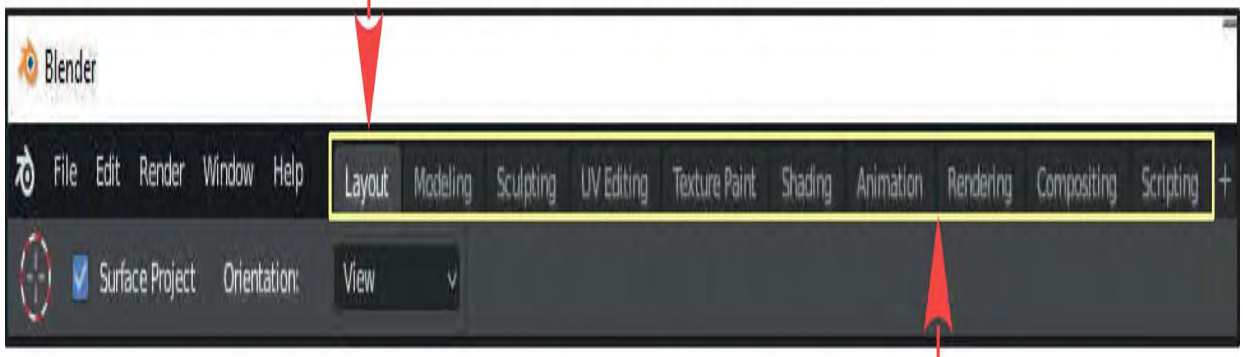
Note: Workbench Render was formerly **OpenGL Render.** The definition has been taken from the Blender Manual. Each Render Engine type displays the view in the computer Screen in different ways **depending on the Viewport Shading method that you select.** This will be explained as you progress through the book.

Workspaces

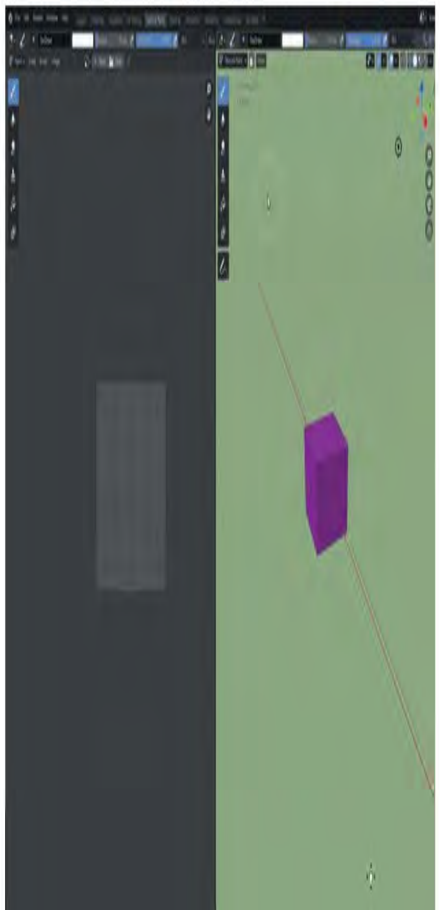
A **Workspace** is the arrangement or configuration of **Editor** panels on the computer Screen. Blender includes numerous Editors for specific functions and these Editors are selected and arranged to facilitate particular operations.

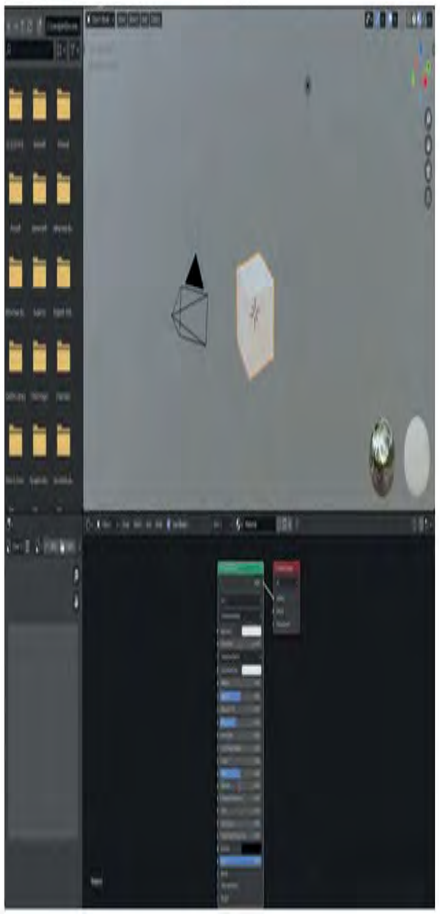
Several Workspaces (Editor arrangements) are provided and may be selected in the Blender Screen Header. There is also the facility for users to build and save specialised arrangements of Editors to suit their working environment for specific tasks.

Layout displays the default **Workspace** in the default Screen Arrangement



Workspace options in the Blender Screen Header



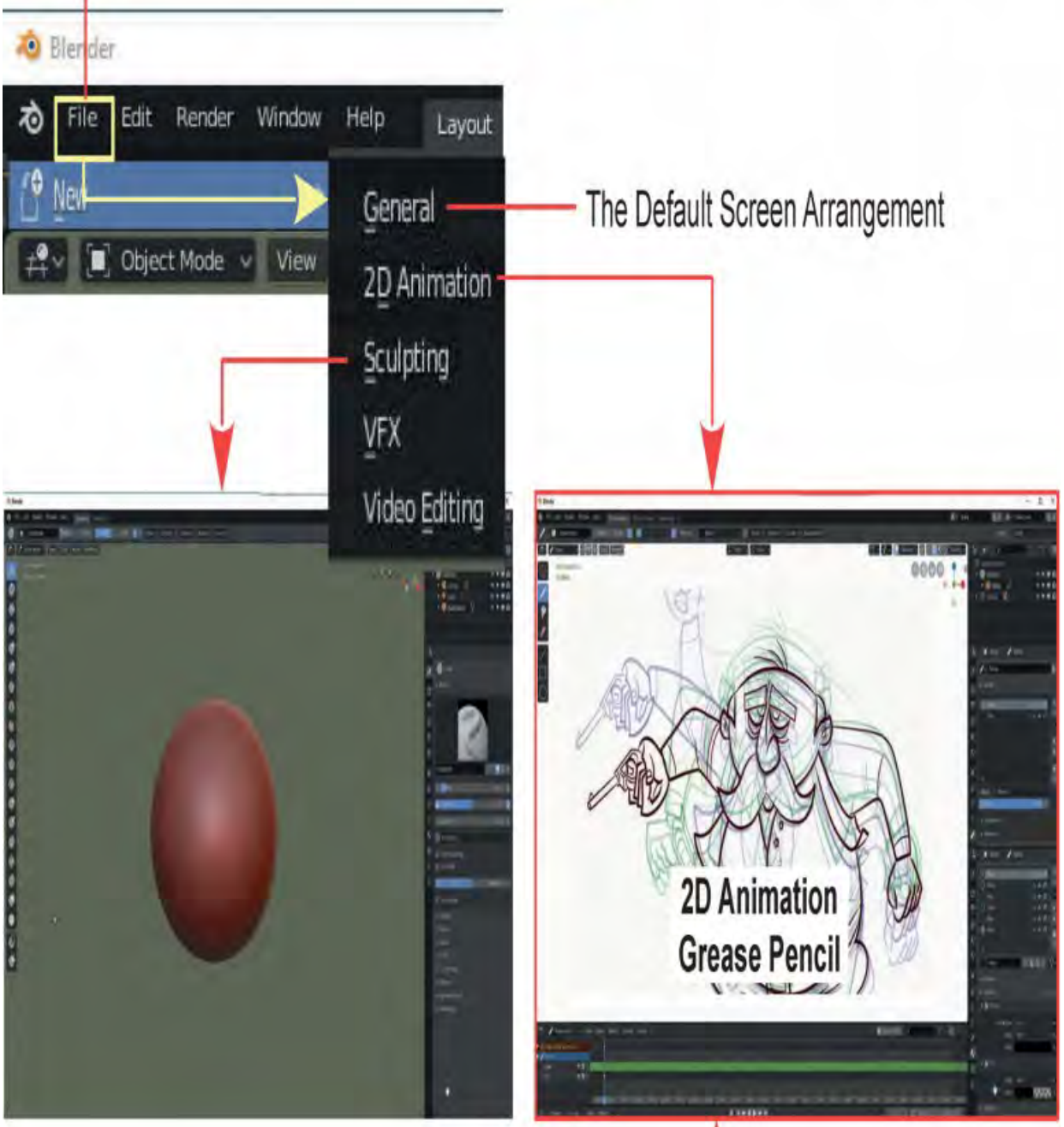


Blender incorporates a dedicated Workspace for creating 2D Animation called **The Grease Pencil.**

This environment provides 2D Animation tools within Blender's 3D Pipeline. 2D Animation creates characters, storyboards, and backgrounds in two-dimensional environments which may be used in advertisements, films, television, computer games, or websites and is just a fun thing to do.

The content and operation of the **Grease Pencil** is covered in a separate dedicated publication titled **The Complete Guide to 2D Animation using the Grease Pencil.** Content and sample pages may be viewed on the Author's website at: https://www.tamarindcreativegraphics.com

Although not designated as Workspaces there are five other Screen arrangements for specific tasks. In the Header at the top of the Screen, click on **File** then **New** to display the option menu.



2D Animation – Grease Pencil

Tamarind Creative Graphics
BLENDER
COMPUTER MODELING & ANIMATION
Blender Graphics
JOHN M. BLAIN
Test Drive Blender
A Starter Manual for New Users
John M. Blain
Blender 2D Animation
Books and Tutorials
Read the Author's Learning Note

1 Understanding the Interface

1.1 Interacting with the Interface
1.2 First Interaction
1.3 Second Interaction
1.4 Getting Help
1.5 Examine the Interface
1.6 Rotating in the 3D Space
1.7 Other Objects
1.8 Using the Outliner Editor
1.9 Working in the 3D Viewport Editor (Object Mode)
1.10 3D Viewport Editor Modes
1.11 Working in the 3D Viewport Editor (Edit Mode)
1.12 Coloring Using the Properties Submenu
1.13 Fire – Quick Method Example
1.14 Animation
1.15 Summary

The Interface

When Blender opens you are presented with a Screen arrangement displaying multiple windows. This arrangement is called the **Graphical User Interface (GUI)** (Figure 1.1). Windows in **Blender** are called **Editors.**

To get you started using Blender, this chapter will show you how to interact with the interface. This introduction will make you familiar with the Editors, entering command using the controls and give you an insight into how the different Editors interrelate. The Preamble preceding this chapter introduced

controls, command instructions and presented example diagrams (**Figures**) touching on a sample of this material.

Of necessity, at this point, very few of the functions available in Blender will be explained in detail. This is just the beginning and it is assumed you are looking at Blender for the first time. Too much information may be confusing. In following chapters the information in Chapter 1 will be repeated and expanded.

By following the instructions with reference to diagrams you will see how Blender works and experience examples. Understanding the Interface and knowing what tools are available, which buttons to press and what to expect is the key to understanding Blender.

1.1 Interacting with the Interface

Before you can explore the Blender interface you have to know the fundamental procedures for entering commands to the program via the Keyboard and Mouse. Refer to the section titled **Formats Conventions and Commands** in the Preamble.

When Blender first opens, the Screen displays the **Graphical User Interface (GUI)** as shown in Figure 1.1. (**Note:** In Figure 1.1 Panels have been delineated with borders and the default background color of the 3D Viewport Editor has been altered).

Blender Interface Header (Upper)
Blender Screen Header
3D Viewport Editor Header
Expansion Button
View Manipulation Widgets
Tool Panel
Lamp
Splash Screen
Cube Object
(Hidden by the Splash
Screen)
Outliner Editor
Camera
Object Properties
Panel
Press N Key to Display
(hidden by default)
Press N Key to Hide
3D Viewport Editor
Properties Editor
Timeline Editor
Blender Interface Header (Lower)

Figure 1.1

The Blender **GUI** opens with the **Splash Screen** panel in the center, showing which version of Blender you have opened (Figure 1.2). There are links included in the **Splash Screen** and buttons which you click to access Screen arrangements for working with specific aspects of the program.

The Splash Screen
(Blender Version 2.80)
Blender Version Number
v 2.80 Release Candidate
blender
cloud.blender.org/spring
New File
General
2D Animation
Sculpting
VFX
Video Editing
Open...
Recover Last Session
Recent Files
Add_GP_Brush.blend
Move_Layers_01.blend
Move_Layers.blend
Multiple_Layers.blend
Layers.blend
Release Notes
Development Fund
Buttons for selecting
Screen Arrangements
Links
3D Viewport Editor
default dark gray color
The Mouse Cursor, positioned to select
the Sculpting Screen Arrangement

Figure 1.2

Note: In Figure 1.1 borders have been added to distinguish the Editors and background colors have been changed. Figure 1.2 shows the Splash Screen for Blender Version 2.80 against the default dark gray background .The Splash Screen for Version 2.81 is different and for Version 3.0.0 different again.

Note: As features are introduced, detailed instruction will be deferred until you have acquired sufficient knowledge to understand the usage. In the early stages of learning Blender this will occur frequently, therefore, a reference will be given to detailed explanation.

1.2 First Interaction

Oops! You clicked the Mouse Button and the **Splash Screen** disappeared?

This is your first interaction with the **GUI**.

Clicking the left Mouse button with the **Mouse Cursor** in the **3D Viewport Editor** cancels the Splash Screen. Should you wish to reinstate it, position the Mouse Cursor over the little black and white Blender logo (button) in the upper left hand corner of the Screen (Figure 1.3).

Position the Mouse Cursor over the **Blender Logo** (highlights blue on Mouse Over)

Figure 1.3

Click the Left Mouse Button (LMB), drag the mouse placing the cursor over **Splash Screen** (highlights blue on Mouse Over) and click LMB again. The Splash Screen is reinstated.

1.3 Second Interaction

When you start something it's a good idea to know how to quit. Blender is no exception. This may be stating the obvious but inevitably things will get messed up and, at some stage, you will want to start over fresh. There are two ways of doing this.

Close the Program and Restart

To close Blender click the **Cross** in the upper right hand corner of the Screen (Figure 1.4). **Click the Cross** means, place the Mouse Cursor over the Cross (Quit button) and click LMB.

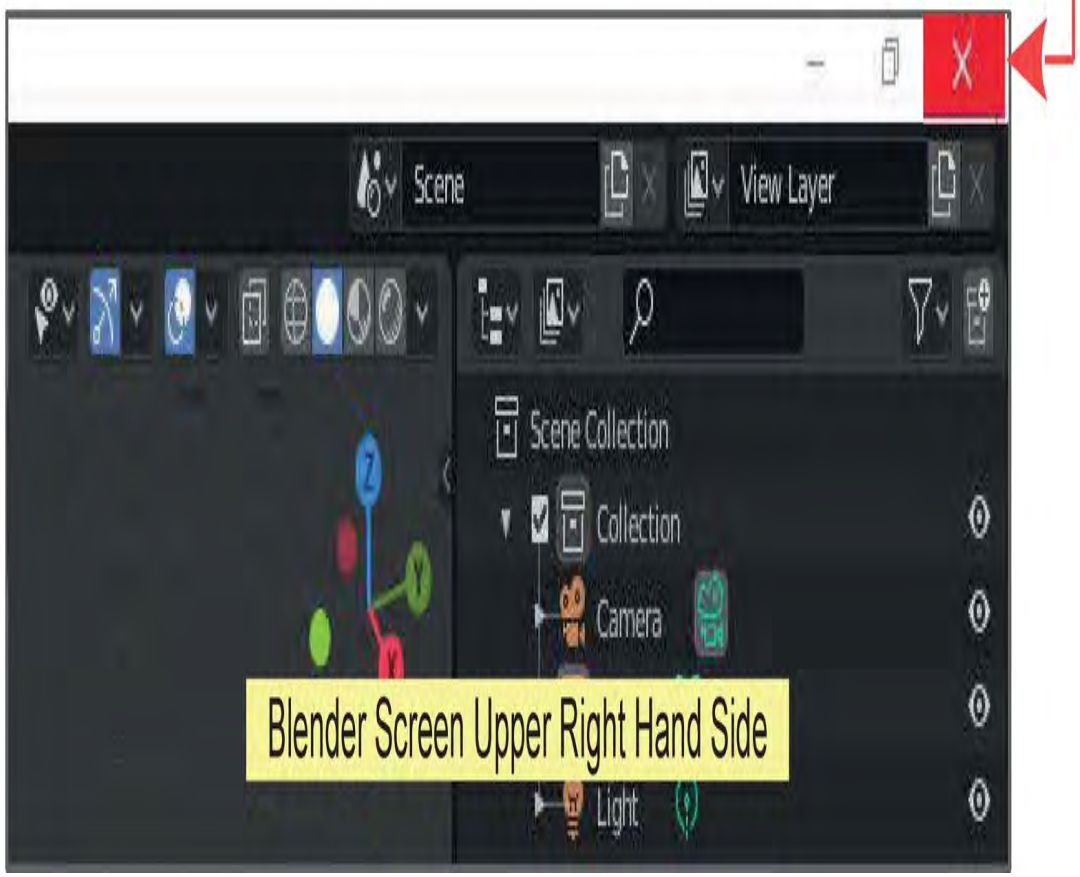


Figure 1.4

Before **Blender** shuts down a panel displays with a reminder to save changes (Figure 1.5). At this point you can Save, Discharge Changes you have made (Don't Save) or Cancel quitting.

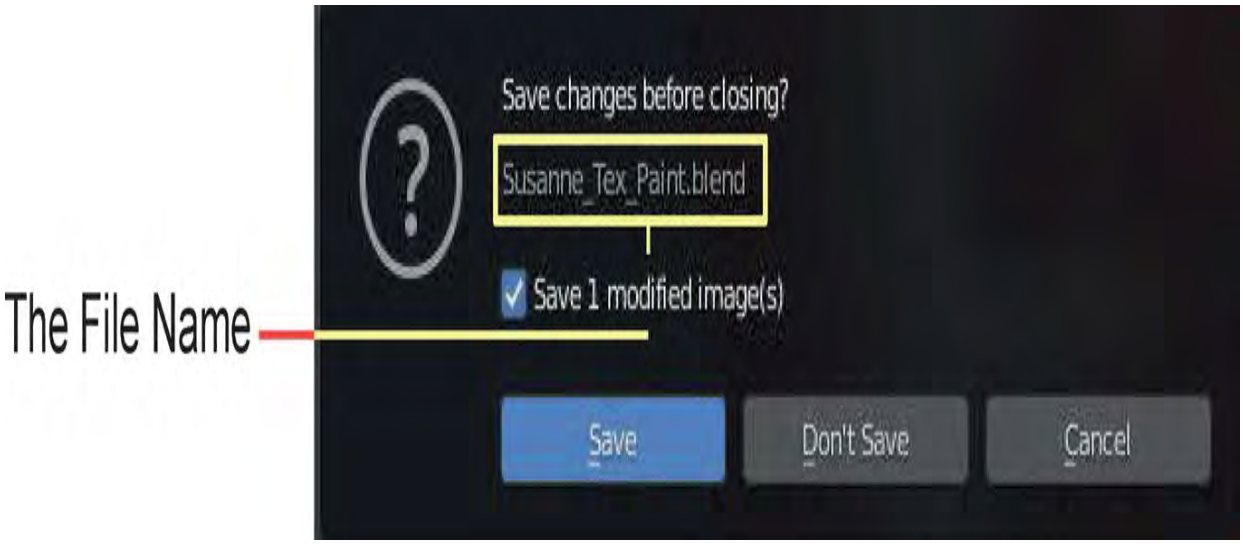


Figure 1.5

Note: When you click the Quit button at the top of the Screen the warning only displays when you have performed an operation. If you haven't done anything the warning does not display.

To Restart the Program double click the Blender Icon on your **Desktop**.

Note the two buttons adjacent to the **Quit button**.

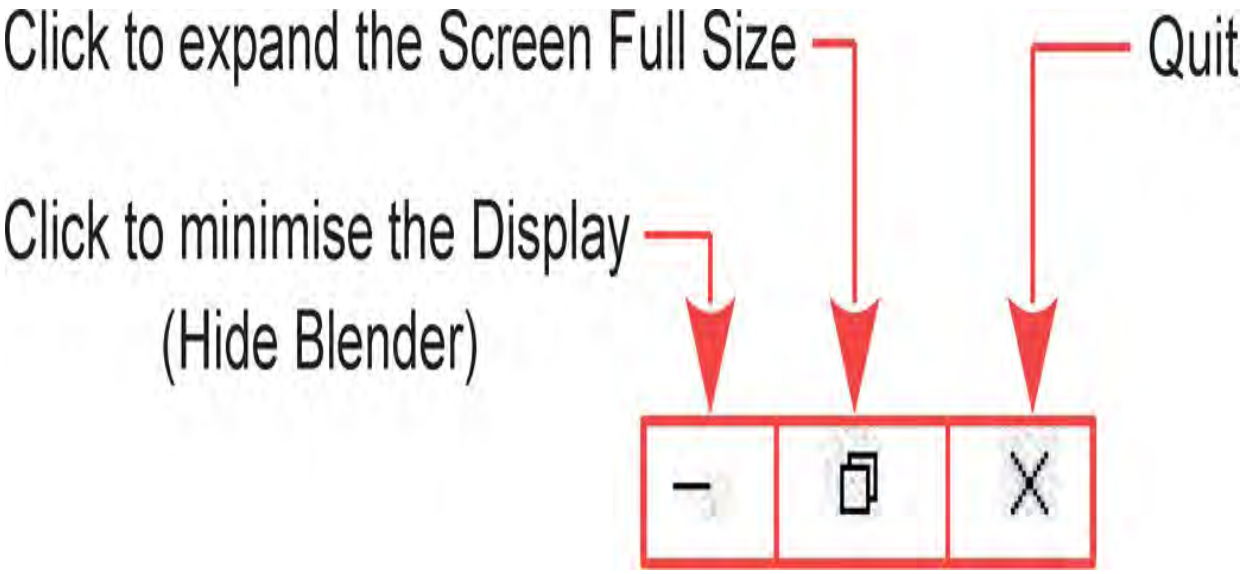


Figure 1.6

Closing the Program - Alternative Method

You may also close **Blender** by clicking on **File** in the upper left hand corner of the Screen and selecting (click) **Quit** in the menu that displays.

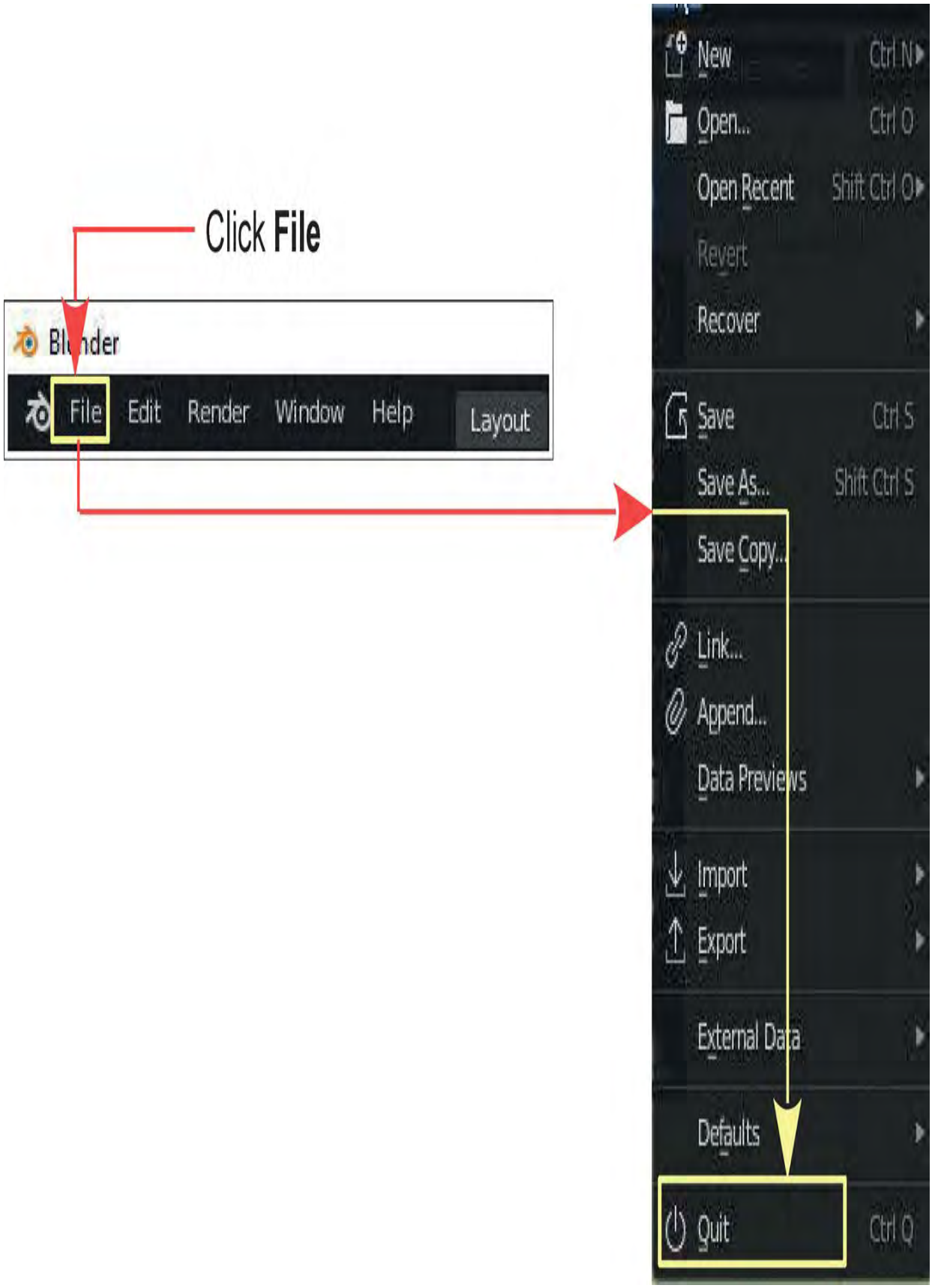

Click File
Blender
File
Edit
Render
Window
Help
Layout
New
Ctrl N
Open...
Ctrl O
Open Recent
Shift Ctrl O
Revert
Recover
Save
Ctrl S
Save As...
Shift Ctrl S
Save Copy...
Link...
Append...
Data Previews
Import
Export
External Data
Defaults
Quit
Ctrl Q

Figure 1.7

Start Over - Restart Without Closing

To restart **without closing** Blender, click (LMB) on **File** in the upper left hand corner of the Screen (highlights blue) (Figure 1.8), drag the Mouse

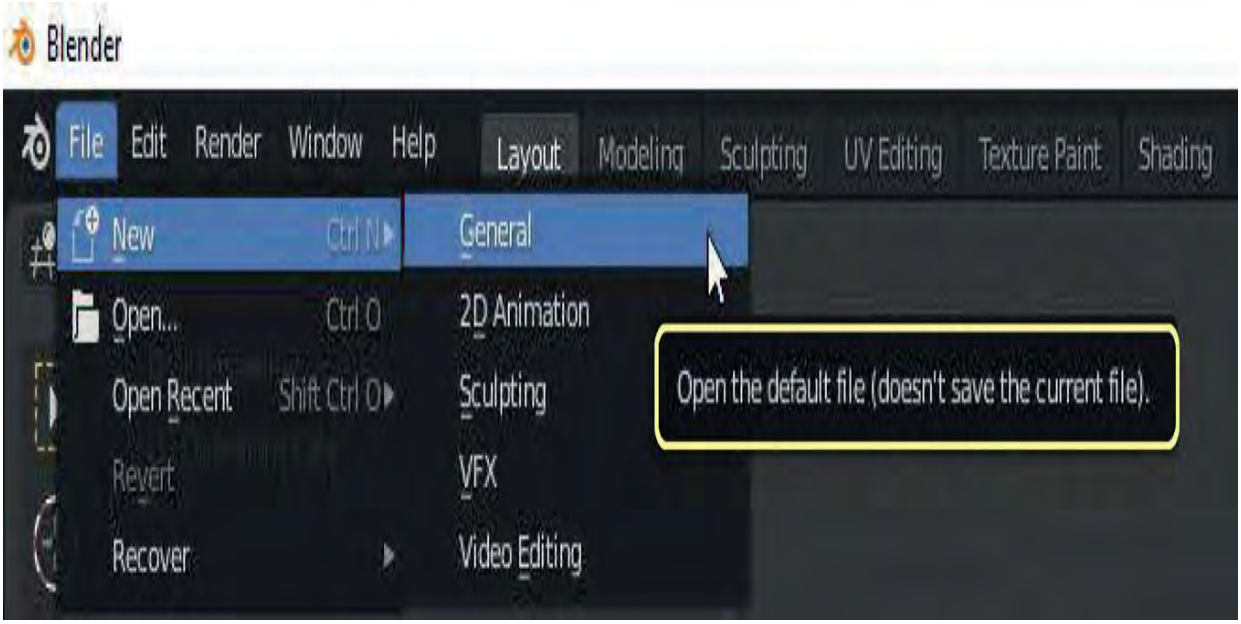


Figure 1.8

This method opens a new default Blender Screen Arrangement.

1.4 Getting Help:

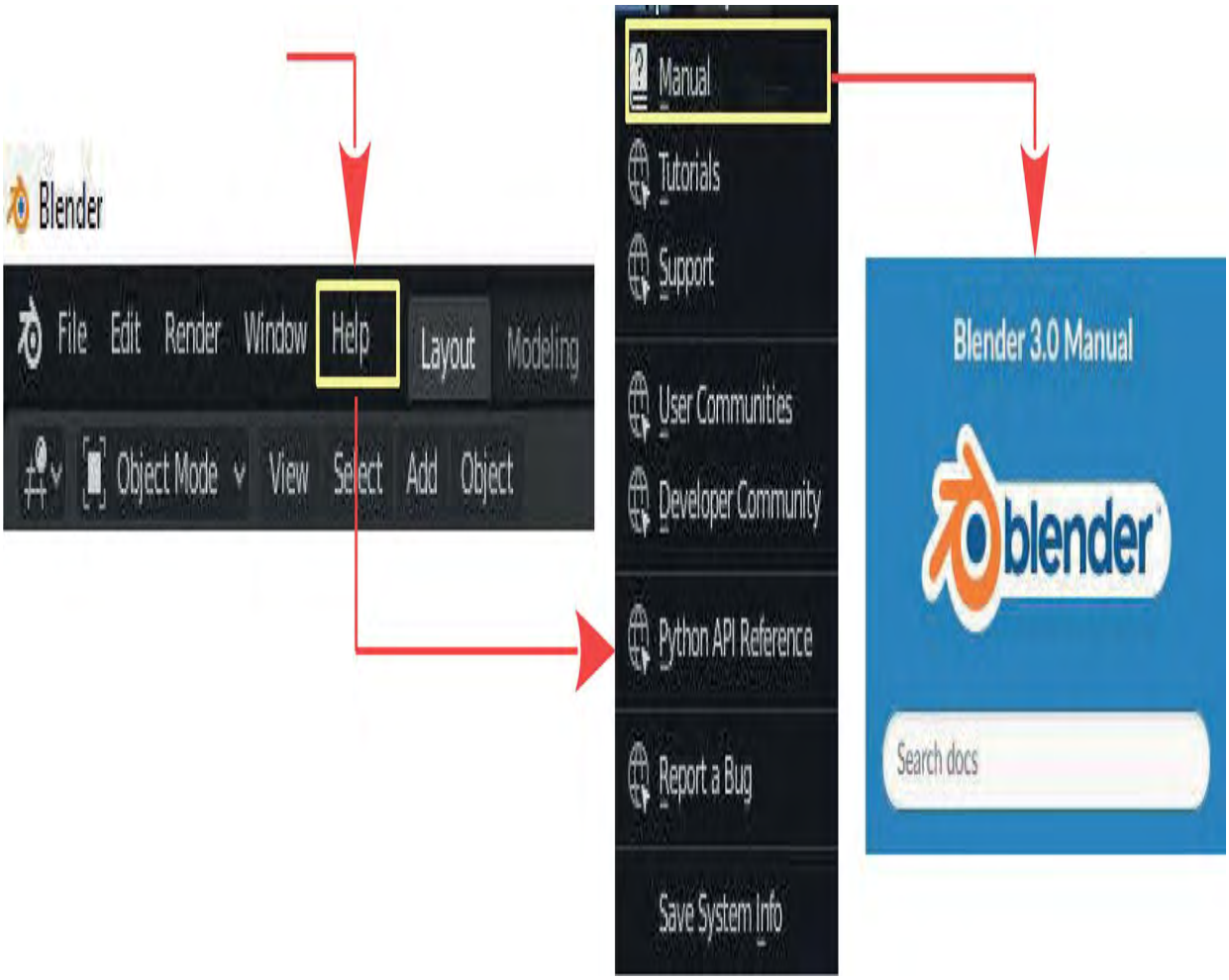


Figure 1.9

If you need help there is always the **Blender Manual**, **Tutorials**, **Support** etc. on the Internet.

1.5 Examine the Interface (Figure 1.1)

Blender Interface Header (Upper)
Blender Screen Header
3D View Editor Header
Expansion Button
View Manipulation Widgets
Tool Panel
Light
Lamp
Splash Screen
Outliner Editor
Cube Object (Hidden)
Camera
Object Properties Panel
Press N Key to Display (hidden by default)
Press N Key to Hide
3D Viewport Editor
3D View Editor
Note: The Editor Background color has been altered
Properties Editor
Timeline Editor
Blender Interface Header (Lower)

Figure 1.10

The **Blender Graphical User Interface** (GUI) is made up with four (4) Panels called **Editors** and three (3) **Headers**. In addition each **Editor Panel** also has its own Header which is usually at the top of the Panel. Headers contain icons and text notations which are buttons for selecting or activating functions.

> Consider **Functions** as the computer code working behind the Scene which make things happen.

When Blender is first opened the Interface displays with the **Splash Screen** in the center of the display in the **3D Viewport Editor** Panel. As you may have discovered, clicking the left Mouse button cancels the Splash Screen leaving the Interface displaying the Blender **Interface Header** (Upper - the white strip across the top of the Screen), the Blender **Screen Header** (the black strip below the Interface Header), and the Blender **Interface Header – Lower** (the strip at the bottom of the Screen). Between the Headers are the four default Editor Panels; the main panel, the **3D Viewport Editor**, the **Outliner Editor** and the **Properties Editor** (at the right hand side of the Screen) and the **Timeline Editor** below the 3D Viewport Editor.

With the Splash Screen removed you see a gray **Cube** in the center of the 3D Viewport Editor.

The 3D Viewport Editor is showing the Cube, suspended in 3D (three dimensional) Space. For convenience the Space is provided with a horizontal **Mid Plane Grid** for reference when positioning Objects in Space. The Grid is marked with the Horizontal X Axis (red line) and the Horizontal Y Axis (green line). By default the Vertical Z Axis is not displayed.

In Blender, the **Cube** is said to be an **Object**. In the default Scene there is also a **Camera** Object and a **Lamp** Object. You see the Objects listed in the **Outliner Editor** (upper RHS of the Screen).

The Cube is one of several basic building blocks from which to commence modeling. The orange outline indicates that the Cube is **selected** which means you can work on it to create something. By clicking **LMB** (left mouse button) with the **Mouse Cursor** over an empty part of the 3D Viewport Editor you will **deselect the Cube**. The orange outline disappears. To re-select the Cube or select another Object you place the Mouse Cursor over the Object and click **LMB**.

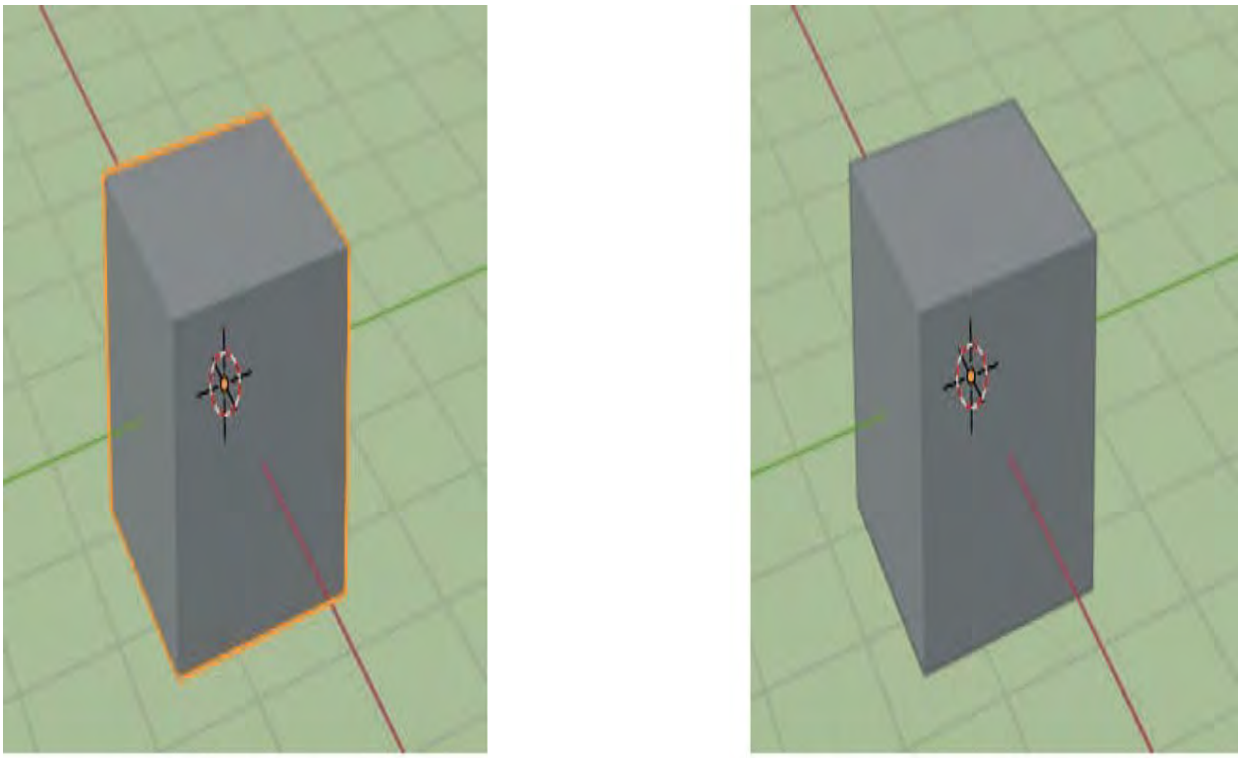


Figure 1.11

Note: In Figure 1.1 the background color of the 3D Viewport Editor has been changed from the default gray hiding the grid. You will be shown how to do this at a later stage (see Chapter 2 – 2.18). Header colors may vary, 3D Viewport and 3D View – Lamp and Light are synonymous.

Note: Default means, that which is displayed or occurs without any action being taken.

1.6 Rotating in the 3D Space

Rotating the Object (Cube)

In the default Scene, The Cube Object is located at the center of the 3D Space. **Have the Cube selected**. Place the Mouse Cursor, in the 3D Viewport Editor to one side of the Cube. Press the **R Key** (rotate) on the Keyboard and drag the Mouse to rotate the Cube. Click LMB to set in position.

Rotating the 3D Space (Viewport)

With the Mouse Cursor positioned in the 3D Viewport Editor, click and hold the **Middle Mouse Button** (Scroll Wheel) and drag the Mouse. The Scene in the 3D View Editor is rotated. Release the Mouse button.

This exercise has distinguished between rotating an Object and rotating the View which is advantageous when modeling. Of course you have messed up the View and may wish to get back to square one. At this stage the options are to restart Blender or restart without closing as described in Section 1.3. There are Keystrokes for rotating the View and Manipulation Widgets for rotating the Object or the View. These will be explained later (see 1.9 and Chapter 2 – 2.7).

1.7 Other Objects

Besides the Cube Object, the default Scene contains a Lamp Object and a Camera Object.

The Lamp represents a light source which illuminates the Scene, determining how you will see the View when it is converted into an Image (see Chapter 15).

The Camera determines what part of the Scene is captured for Rendering (Chapter 15 – 15.3).

The Lamp and the Camera have Properties which may be modified to produce effects. How to do this will be placed on hold for the time being.

There are many Object which you can add to the Scene. With one exception the Objects are not preconstructed models of characters but mesh shapes (Primitives) from which you begin modeling.

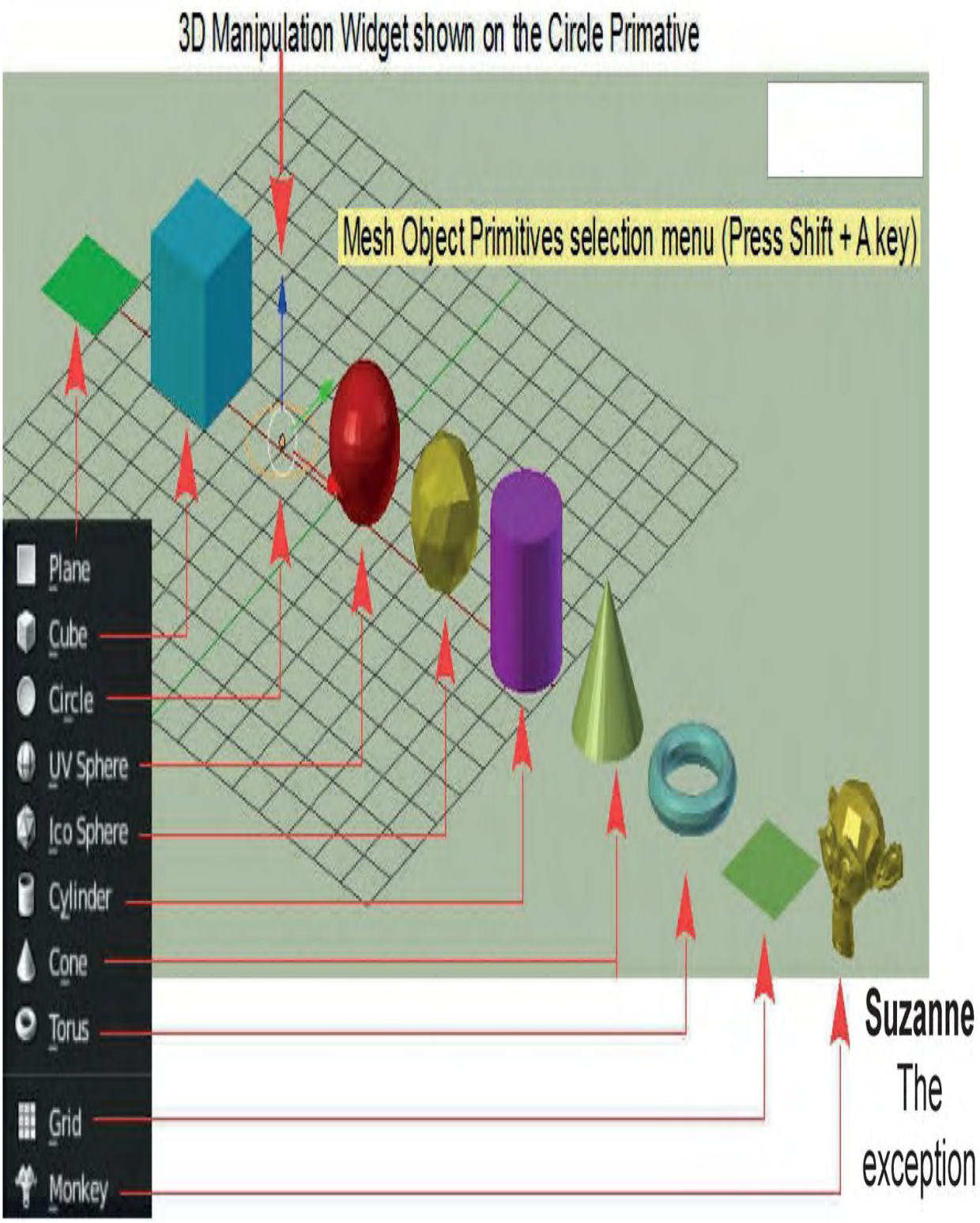

3D Manipulation Widget shown on the Circle Primative
Mesh Object Primitives selection menu (Press Shift + A key)
Plane
Cube
Circle
UV Sphere
Ico Sphere
Cylinder
Cone
Torus
Grid
Monkey
Suzanne
The
exception

Figure 1.12

To demonstrate adding an Object, the one exception, the preconstructed Mesh model of a Monkey, affectionately named **Suzanne**, will be added to the default Scene.

The basic principle in Blender, when modeling, is to add one of the basic Mesh Objects, sometimes called Primitives, then modify (model) that shape into whatever you want. You may combine several primitives if you wish.

To Add a new Object you either click the **Add button** in the 3D Viewport Editor Header or with the Mouse Cursor in the 3D Viewport Editor, press the **Shift Key + the A Key** (keyboard) to display the **Add Menu**.

To add **Suzanne** (the Monkey Object), mouse over on **Mesh** then in the Mesh sub menu click **Monkey**.

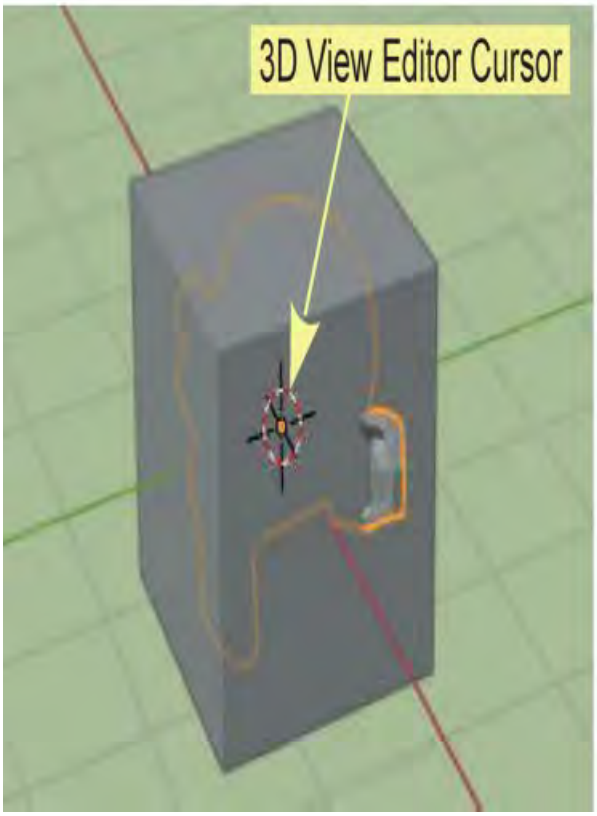


Figure 1.13

To add **Suzanne** (the Monkey Object), mouse over on **Mesh** then in the Mesh sub menu click **Monkey**.

All New Objects added into a Scene are entered at the position of the **3D Viewport Editor Cursor**. **Yes! There is another Cursor**. By default the 3D Viewport Editor Cursor is located at the center of the 3D Space. The default Cube Object is also located at the same position and when Suzanne (Monkey) is entered she is also located at this position.

When a new Object is added to the Scene, the new Object becomes the selected Object, hence you see the orange outline and in this particular case you see part of Suzanne's Ear protruding from the surface of the Cube. To see Suzanne clearly hide the Cube using the **Outliner Editor**.

1.8 Using the Outliner Editor

Figure 1.14

The **Outliner Editor** lists all the Objects in the Scene and provides a means of organising Objects into groups. This is a great help when compiling complicated Scenes.

In Figure 1.14 you see the Objects in the Scene listed in a group named **Collection**.

How to organise Collections will be explained in Chapter 12 but for the moment, click on **Cube**, then click on the little **Eye Icon** adjacent to Cube (Figure 1.15) to hide the Cube in the Scene. This does not delete the Cube, it merely cancels the display. You click the eye icon a second time to reinstate the Cube in the 3D Viewport Editor. If you click the eye icon adjacent to Collection the display of every Object in the Collection is cancelled.

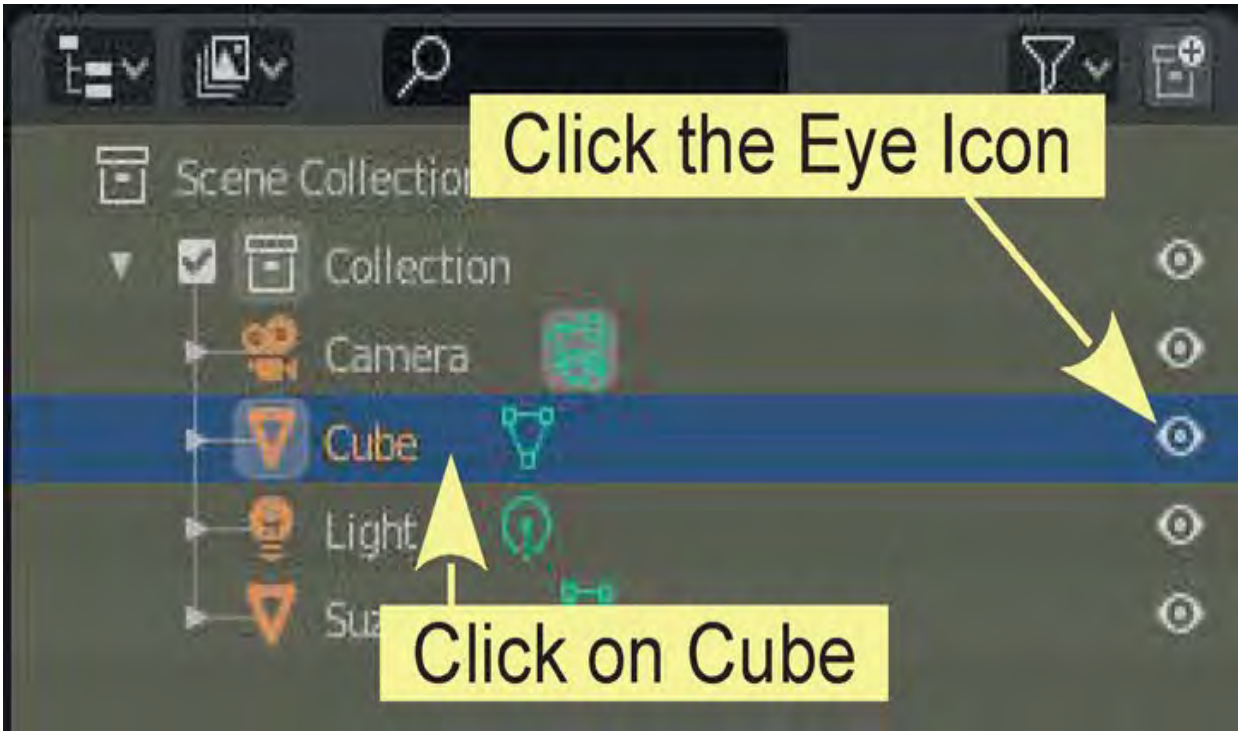


Figure 1.15

1.9 Working in the 3D Viewport Editor (Object Mode)

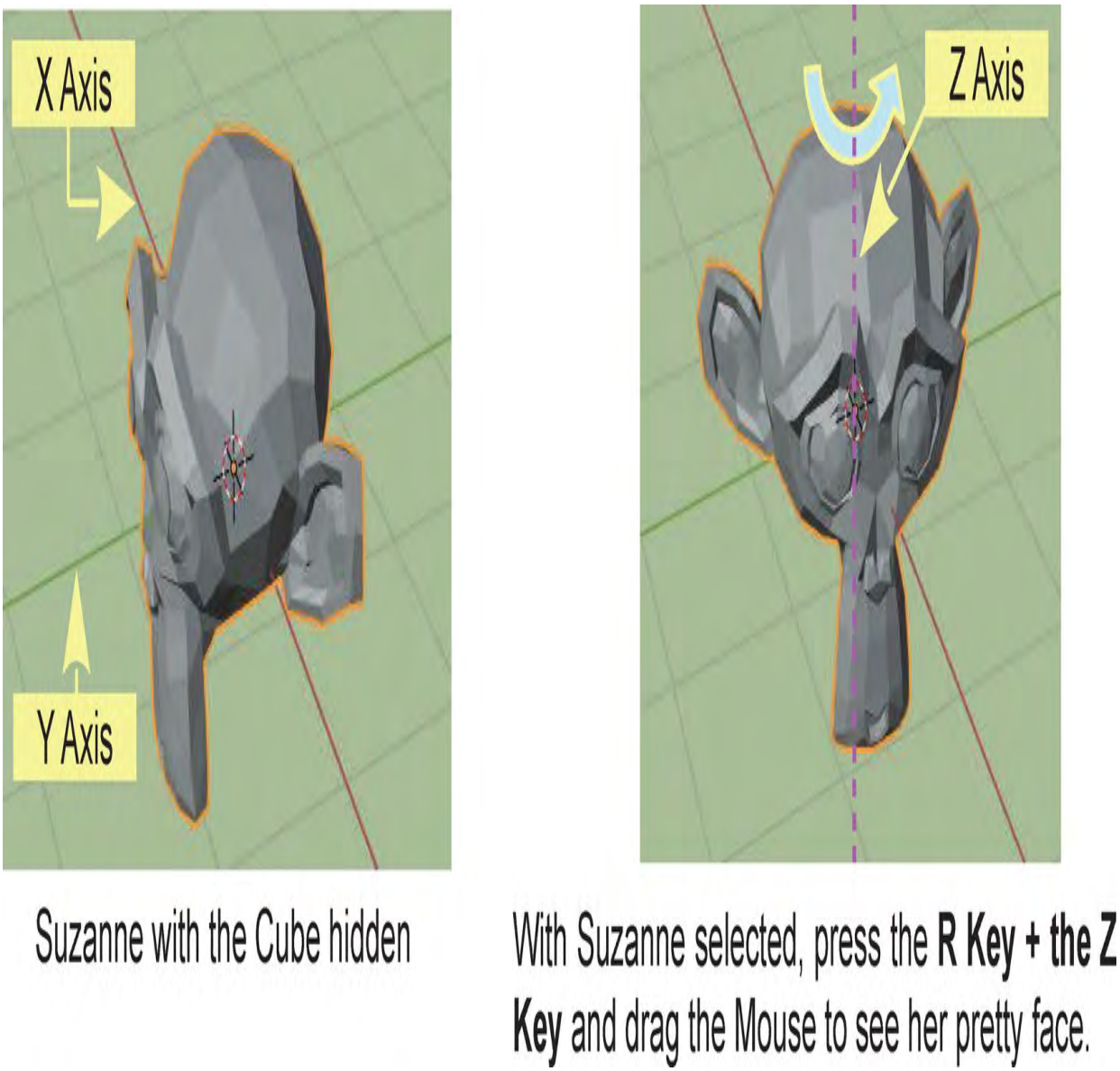


Figure 1.16

Suzanne with the Cube hidden

With Suzanne selected, press the **R Key + the Z Key** and drag the Mouse to see her pretty face.

Rotation: R Key + Z Key means, Rotate about the **Vertical Z Axis.**

There are many functions which can be employed in the 3D Viewport Editor. Rotating an Object is one. You also **Scale** the selected Object (Press the S Key and drag the Mouse) (S Key + X confines the Scale operation to the

X Axis). You **Translate** (move and reposition) the Object in the 3D Viewport Editor (Press the G Key (G for Grab) and drag the Mouse) (G Key + Y confines the movement to the Y Axis).

The function of the Keystrokes are replicated by **Widgets** which are activated from the **Tool**

The function of the Keystrokes are replicated by **Widgets** which are activated from the **Tool Panel in the 3D Viewport Editor**.

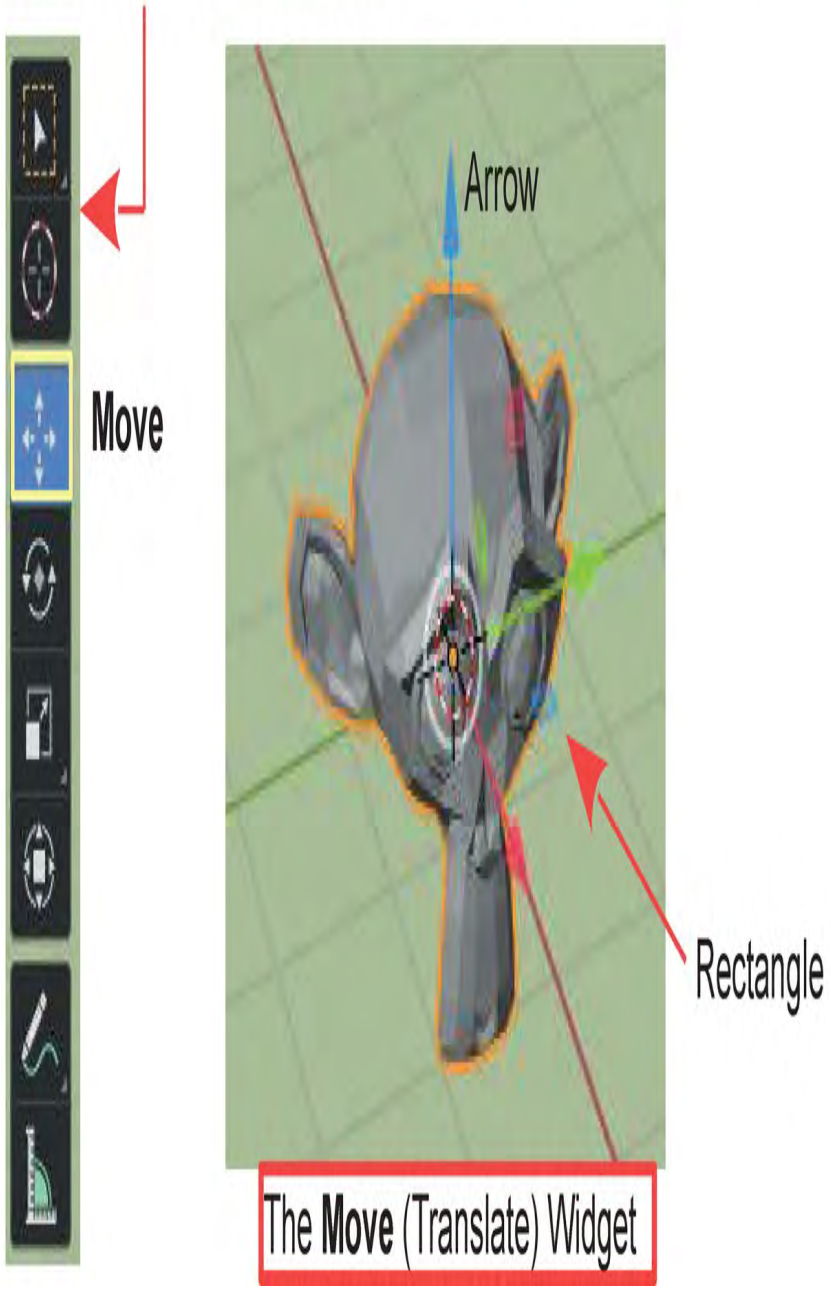


Figure 1.17

The colored **arrows** and **rectangles** are **Widget Control Handles** which you click (LMB), hold and drag to Translate (move) the selected Object in the

Scene. Clicking an arrow confines the movement to an Axis in the Scene. Clicking a rectangles confines the movement to either the X,Y or Z Plane.

With the Cube hidden, using the Transform Widget, move Suzanne to one side of the center of the Scene, then in the Outliner Editor, click the eye icon adjacent to Cube to reinstate the display of the Cube in the 3D Viewport Editor.

1.10 3D Viewport Editor Modes

The 3D Viewport Editor can be displayed in different Modes depending on what operation is to be performed.

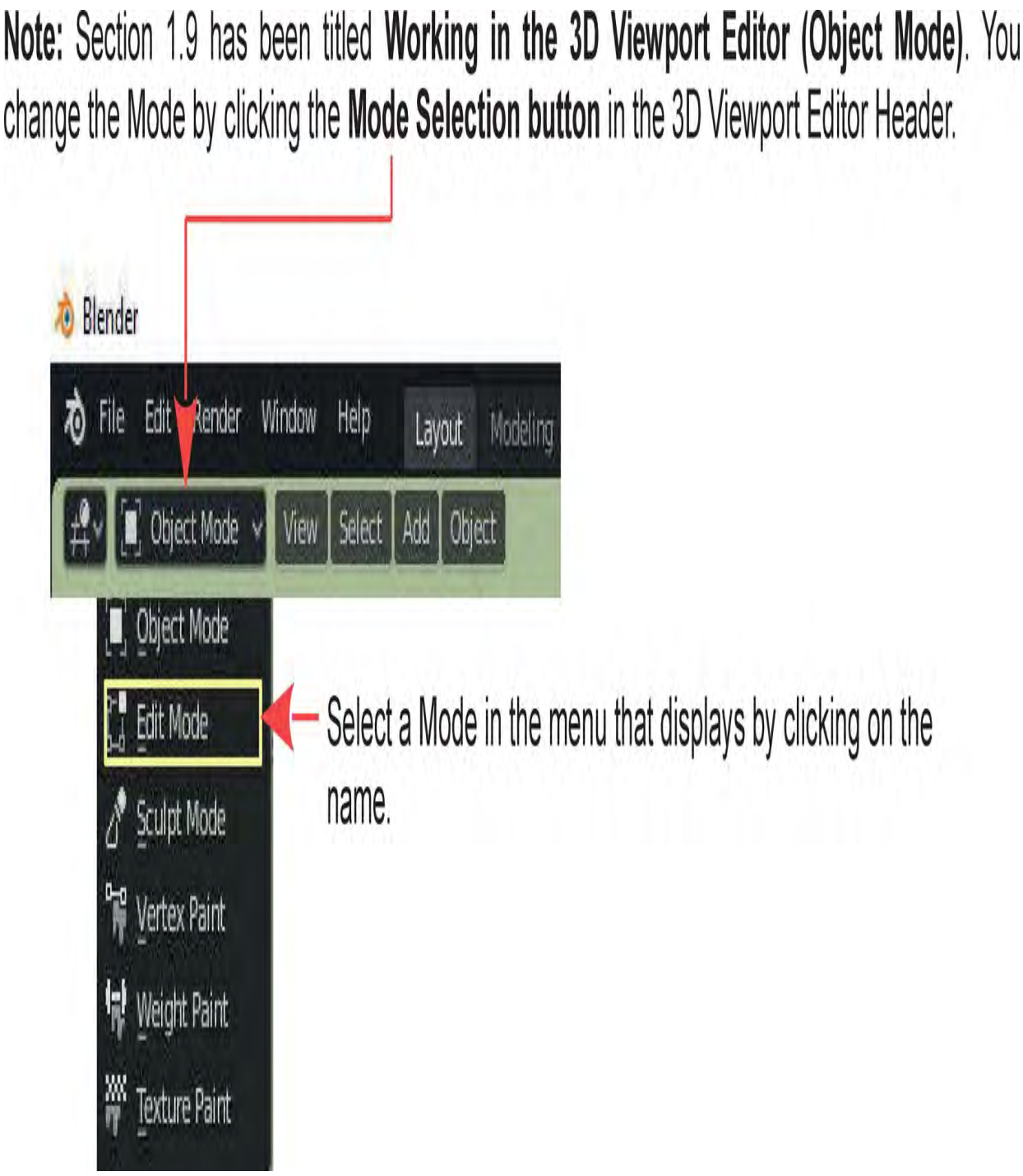


Figure 1.18

1.11 Working in the 3D Viewport Editor (Edit Mode)

Note: You will find there are many times when you want to switch between Object Mode and Edit Mode. Instead of using the Mode Selection Menu

simply press the **Tab Key** on the Keyboard to **Toggle** between the two Modes (have the Object selected).

In **Edit Mode** the default Cube displays with its **Faces** a different shade of gray, with orange dots at each corner called **Vertices**, connected by thin orange lines which are the **Edges**.

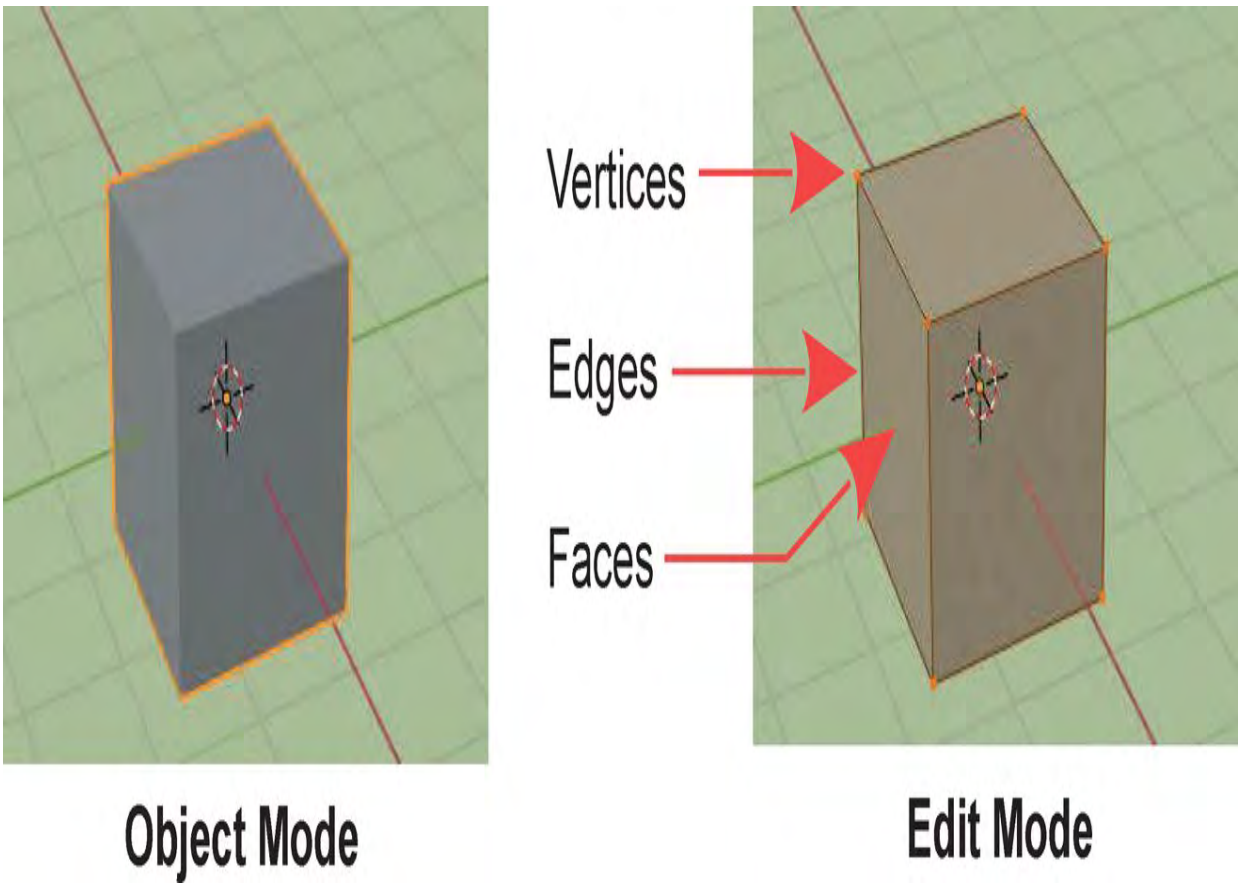


Figure 1.19

When changing from Object Mode to Edit Mode all **Vertices**, **Edges** and **Faces** are selected.

Click **LMB** in the 3D Viewport Editor to deselect all the Vertices, Edges and Faces.

Note: In Edit Mode, the Mouse Cursor displays as a **white cross** +. Figure 1.20

Figure 1.20

Position the Mouse Cursor (white cross) over one of the Vertices and click LMB to select a Vertex on the Cube (vertex turns white).

Note: Selection in 3D Viewport Editor, in Edit Mode, is by default, set to **Vertex Select.**

You change the **Selection Mode** in the 3D Viewport Editor Header.

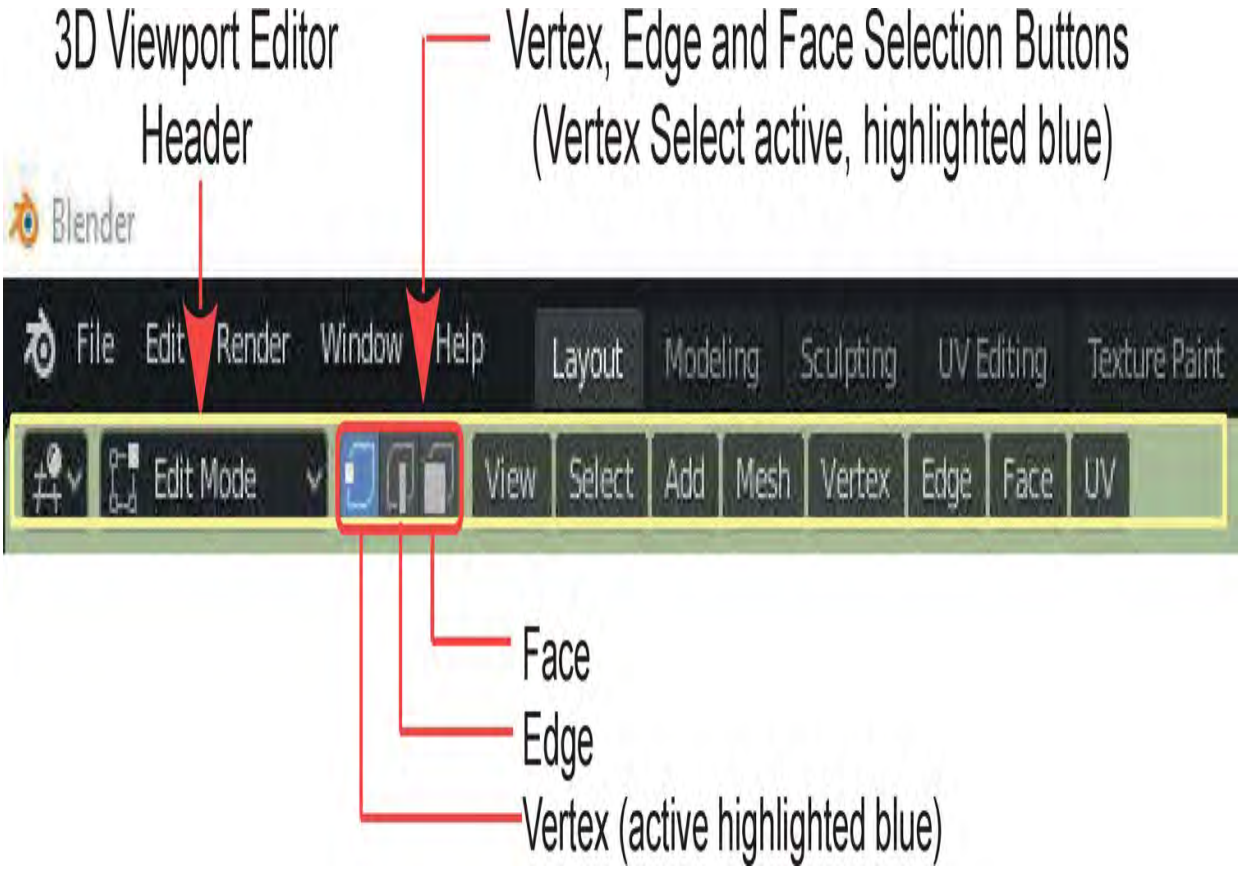


Figure 1.21

Note: When changing to Edit Mode the array of buttons in the 3D Viewport Editor Header is different to Object Mode and the options in the Tool Panel change. You are able to select a Vertex since the Vertex Select button in the Header is active (highlighted blue). You have the options to select Edges or Faces.

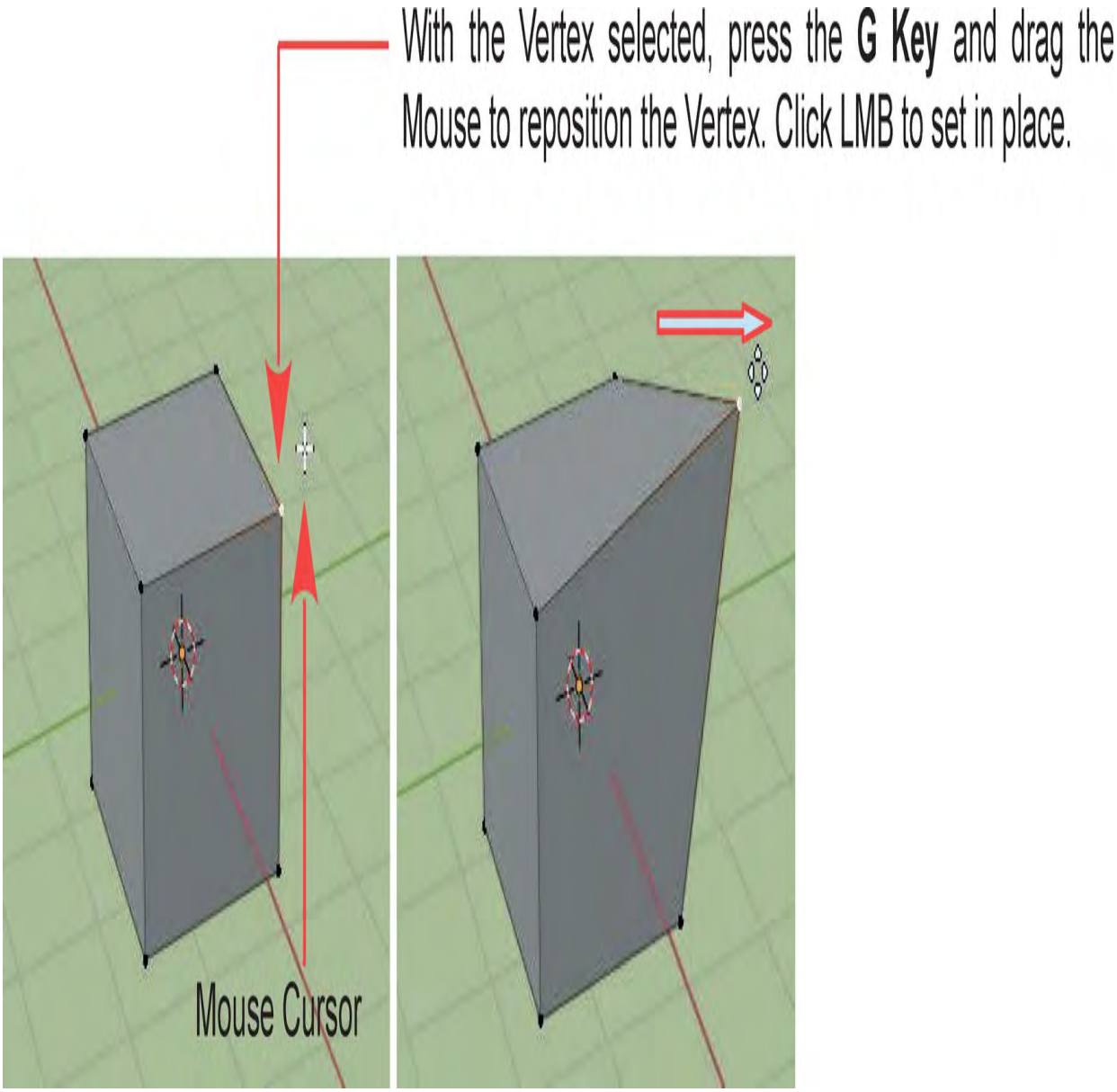


Figure 1.22

This simple rudimentary procedure is the basis of **Modeling**.

1.12 Coloring Using the Properties Editor

Bear in mind, the objective in this chapter is to familiarise you with the Graphical User Interface, not to explain all the functions available in Blender. You have been introduced to Editors and Headers and given a small insight into operating in 3D Space and working with Objects.

The **Properties Editor** contains controls which affect what happens in the 3D Viewport Editor. Some controls apply to the Scene as a whole and some specifically affect the properties of the Selected Object. There are usually multiple Objects in a Scene. The controls in the Properties Editor change when an Object is selected and affect only the Selected Object.

To demonstrate, it will be assumed, you have a Scene containing a Cube Object and a Monkey Object. In this case the Cube has been modified per the previous example (Figure 1.22) and Suzanne has been moved to the side (Figure 1.23).

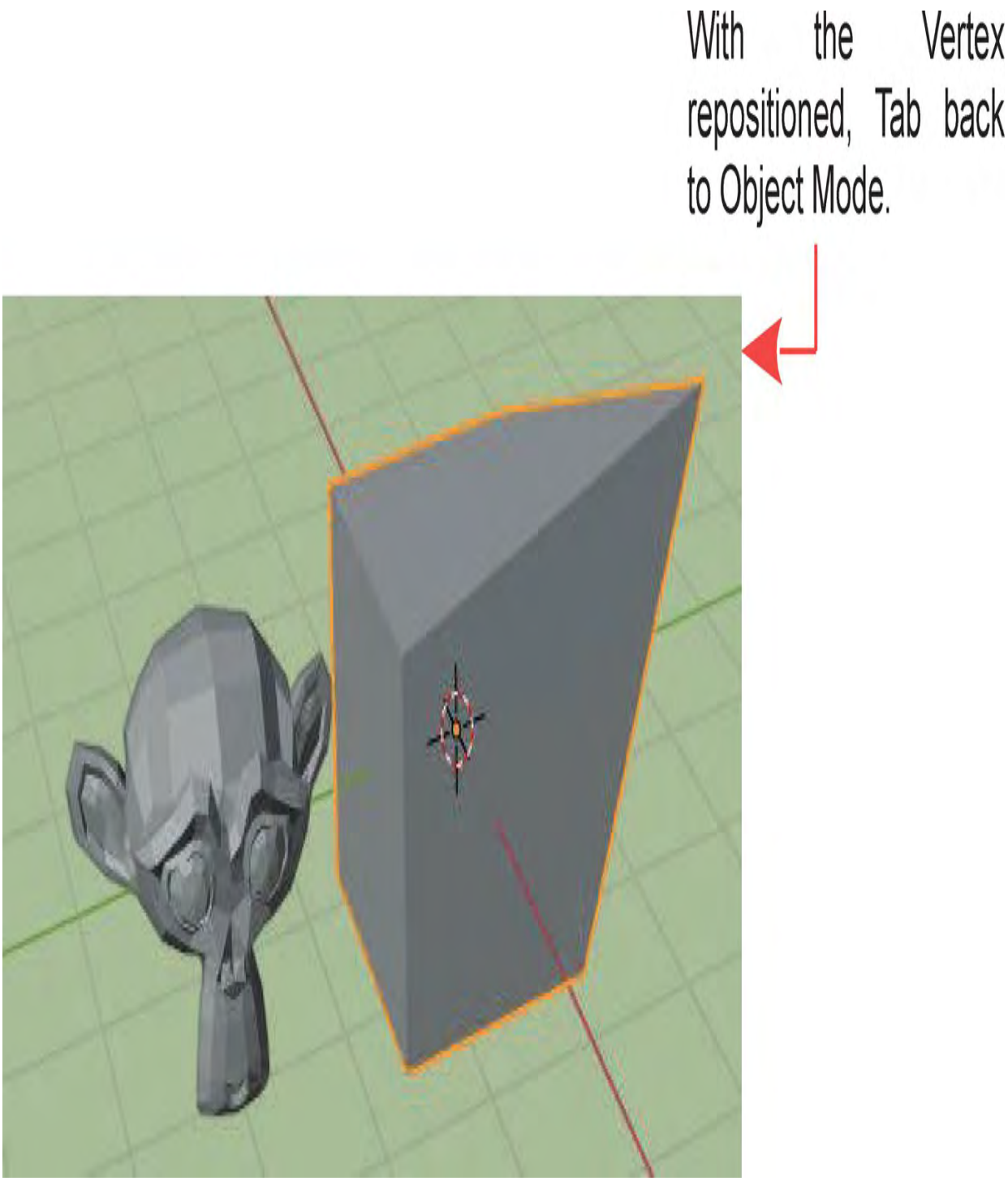


Figure 1.23

The Properties Editor will be used to add a color to Susanne. **With Suzanne selected**, click on the **Material Properties** button at the left hand side of the **Properties Editor** (Figure 1.24). Remember, Monkey is called

Suzanne. With Suzanne selected the Material button controls only affect Suzanne. The Material button controls for Suzanne at this stage consist of the single **New Button**. This is telling you that Suzanne has no Material applied. At this point consider **Material** to mean **Color**.

The vertical array of buttons at the left hand side of the Properties Editor will be explained in due course but for the moment only be concerned with coloring Suzanne.

The Material Properties Button

Figure 1.24

Why Color?

Color adds a new dimension to a Scene. Objects become more interesting and color in a Scene creates atmosphere which enhances the story or message being told.

When constructing a Scene you should consider having a plan in mind. In the following demonstration the plan will be to have Suzanne appear blue with cold, sitting somewhere in the background. The deformed Cube will be set on fire. Suzanne will move forward, changing color as she warms up. Having Susanne colored makes the Scene a little more interesting. Besides, it introduces coloring controls and demonstrates the interrelation between the 3D Viewport Editor and the Properties Editor.

This demonstration will be a very simple unsophisticated example of animation which will give a preview of a physical simulation (Fire and Smoke).

To add **Material** (color) to Suzanne (the selected Object) click the **New Button** in the Properties Editor. The display expands (Figure 1.25).

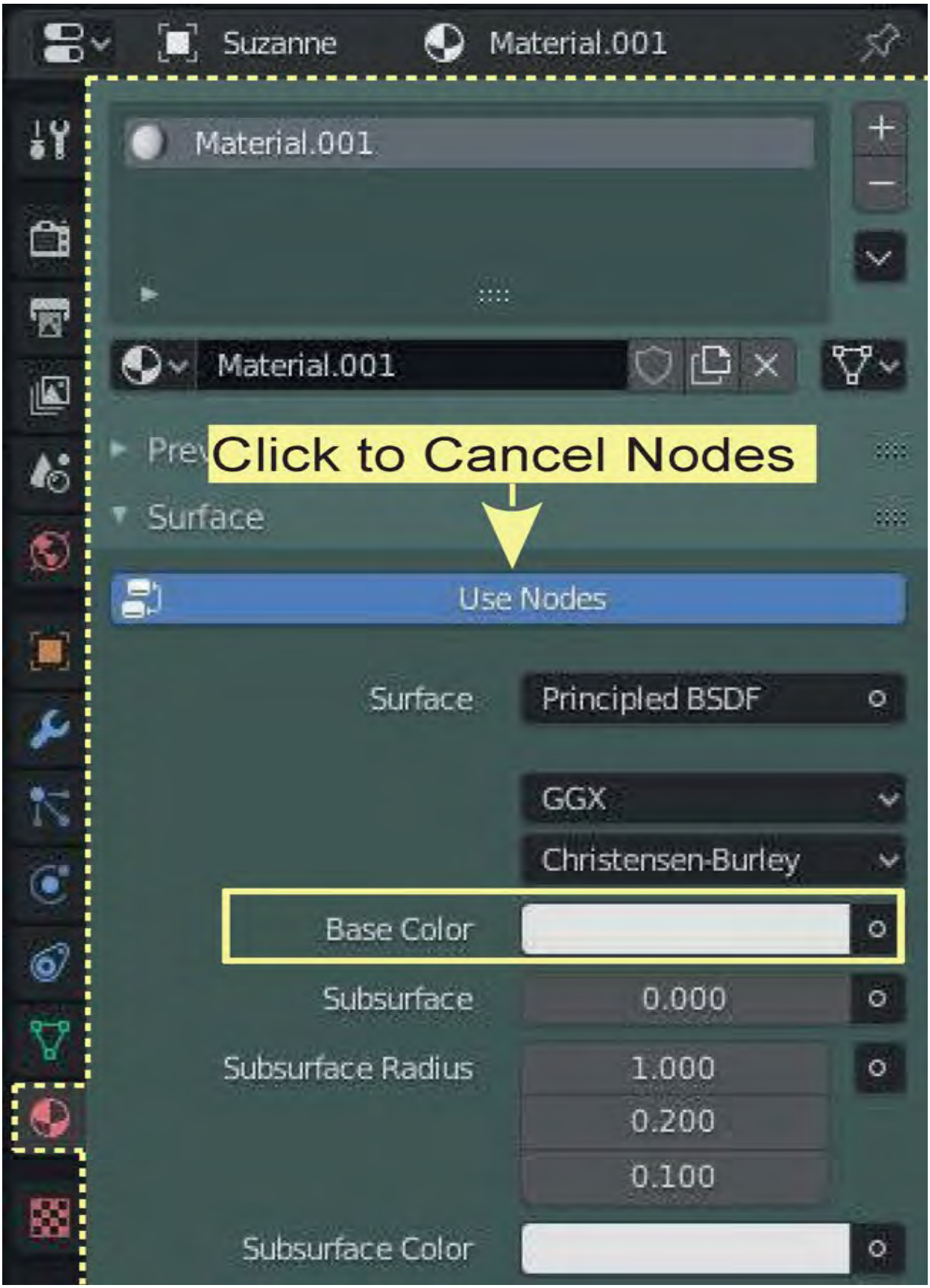

Suzanne
Material.001
Material.001
Material.001
Click to Cancel Nodes
Surface
Use Nodes
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
1.000
0.200
0.100
Subsurface Color

Figure 1.25

By default, Material Color is being controlled by a hidden **Node System**. The blue bar with the notation **Use Nodes** indicates that the Node System is in control.

Nodes are explained in Chapter16, but for the moment click on the blue **Use Nodes** button to **cancel the Node System**.

With the Node System cancelled the Properties Editor, Material buttons display with a reduced number of functions (Figure 1.26 **Note:** diagrams are curtailed to save page space).

Remember the Material Properties button controls only apply to the selected Object in the 3D Viewport Editor, in this case Suzanne.

To turn Suzanne blue, click on the white **Base Color** bar to display a **Color Picker**. The Color Picker gives the options to select one of three color schemes; **RGB**, **HSV** or **Hex**. By default the HSV color scheme is selected. Click on the **RGB** button to change the color scheme to RGB (**R**ed, **G**reen, **B**lue).

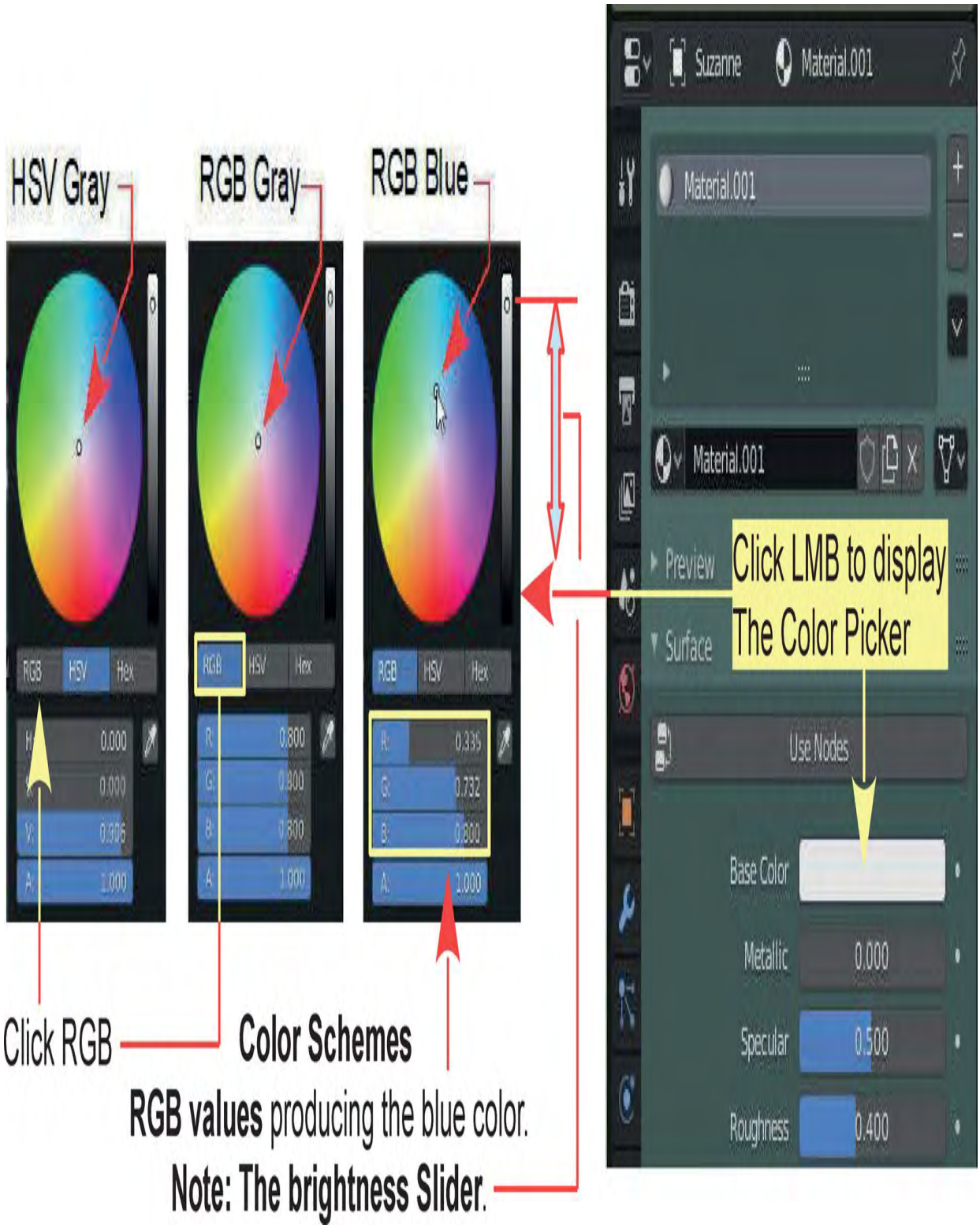

HSV Gray
RGB Gray
RGB Blue
Suzanne
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Click LMB to display
The Color Picker
Base Color
Metallic
0.000
Specular
0.500
Roughness
0.400
RGB
HSV
Hex
Click RGB
Color Schemes
RGB values producing the blue color.
Note: The brightness Slider.

Figure 1.26

When you add a Color to an Object the RGB values are by default, R: 0.800, G: 0.800, B: 0.800 and A: 1.000. The RGB values generate the Blender default gray color, while the A (Alpha) value sets the transparency of the color (A: 1.000 = Opaque, A: 0.000 = Fully Transparent).

To make Suzanne blue in the face, click LMB in the RGB Color Picker Circle where it displays blue. To select the exact blue shown in the diagrams you may double click on a numeric value (displays a typing cursor) hit delete or backspace and retype a new value (Figure 1.27 opposite).

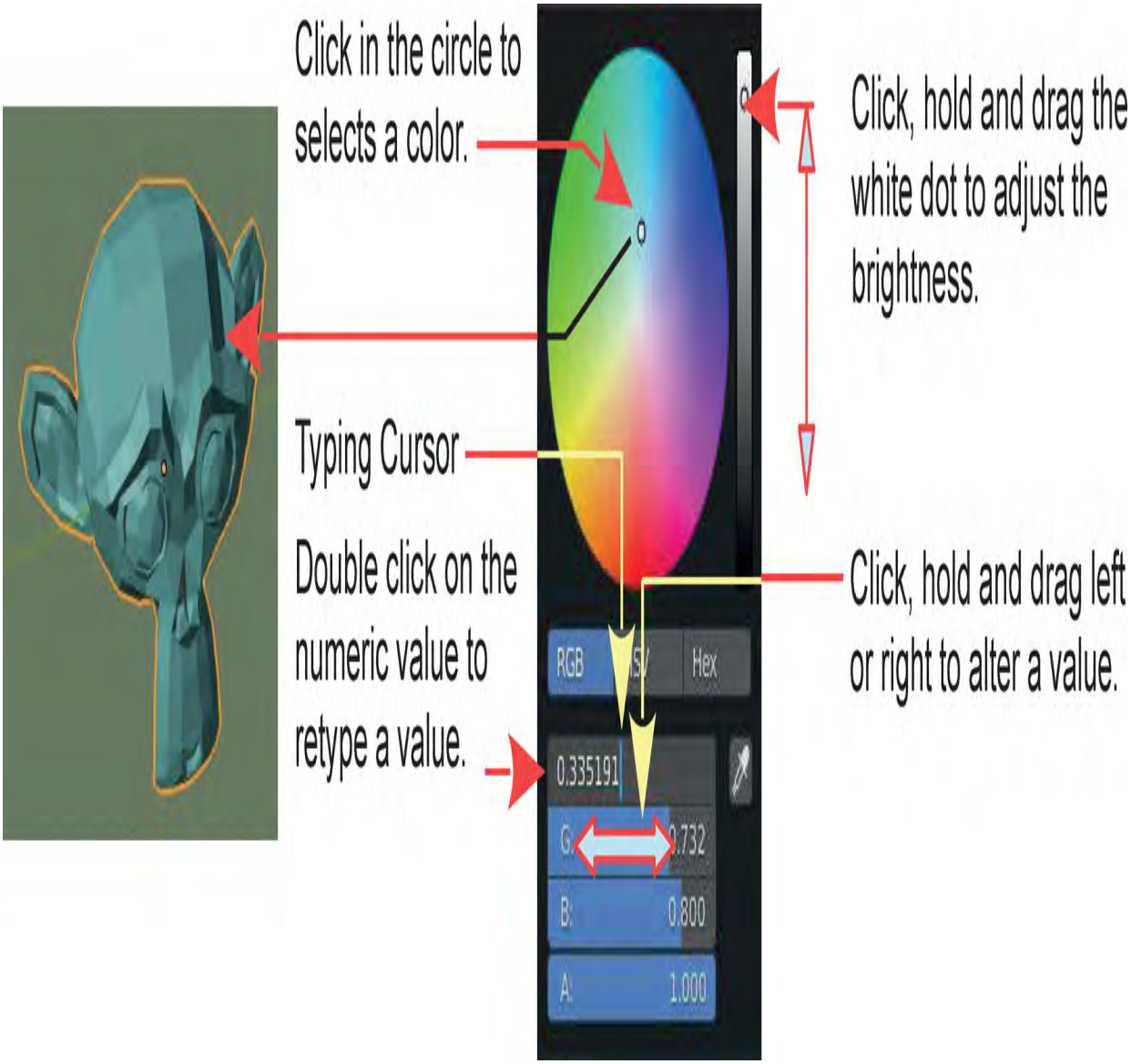


Figure 1.27

When you select a color in the circle or alter the numeric values, the color of the selected Object changes. **Note:** This is a basic procedure demonstrating the application of Material color prior to understanding **Viewport Shading Modes** and **Material Nodes** (see Chapter 14 and 16).

Suzanne is now shivering and blue in the face. Move Suzanne towards the back of the Scene. To warm Susanne up start a fire.

1.13 Fire - Quick Method Example

Blender allows you to create effects from first principles. This applies to modeling and creating physical effects. There are several Quick effects one of which is a method for creating a fire effect. Make note this is a **Quick Method** which has a limited application. You will be instructed in how to generate this effect from first principles later on and when you understand the procedures the results are limitless.

To start a fire you need something to burn so let's set the Cube that has been modified alight.

Deselect Suzanne and select the modified Cube Object. You may use the default Cube.

Note: The Cube doesn't have to be modified and you can use any Mesh Object. If you have been following along, the modified Cube will already be in the Scene, therefore, its convenient to use for the demonstration.

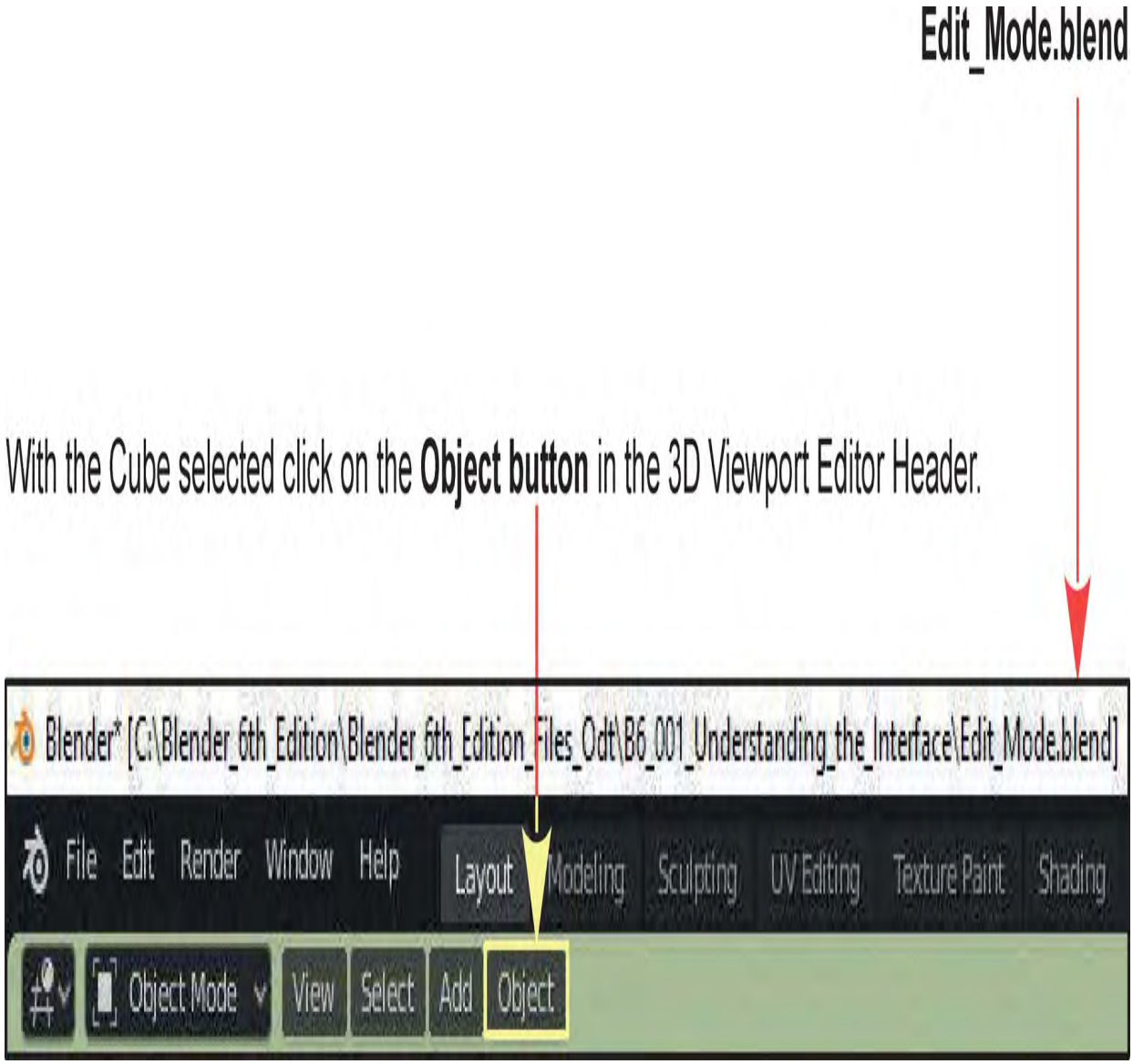


Figure 1.28

Clicking the Object button displays a selection menu. Towards the bottom of the menu, click **Quick Effects** and in the sub menu select **Quick Smoke**.

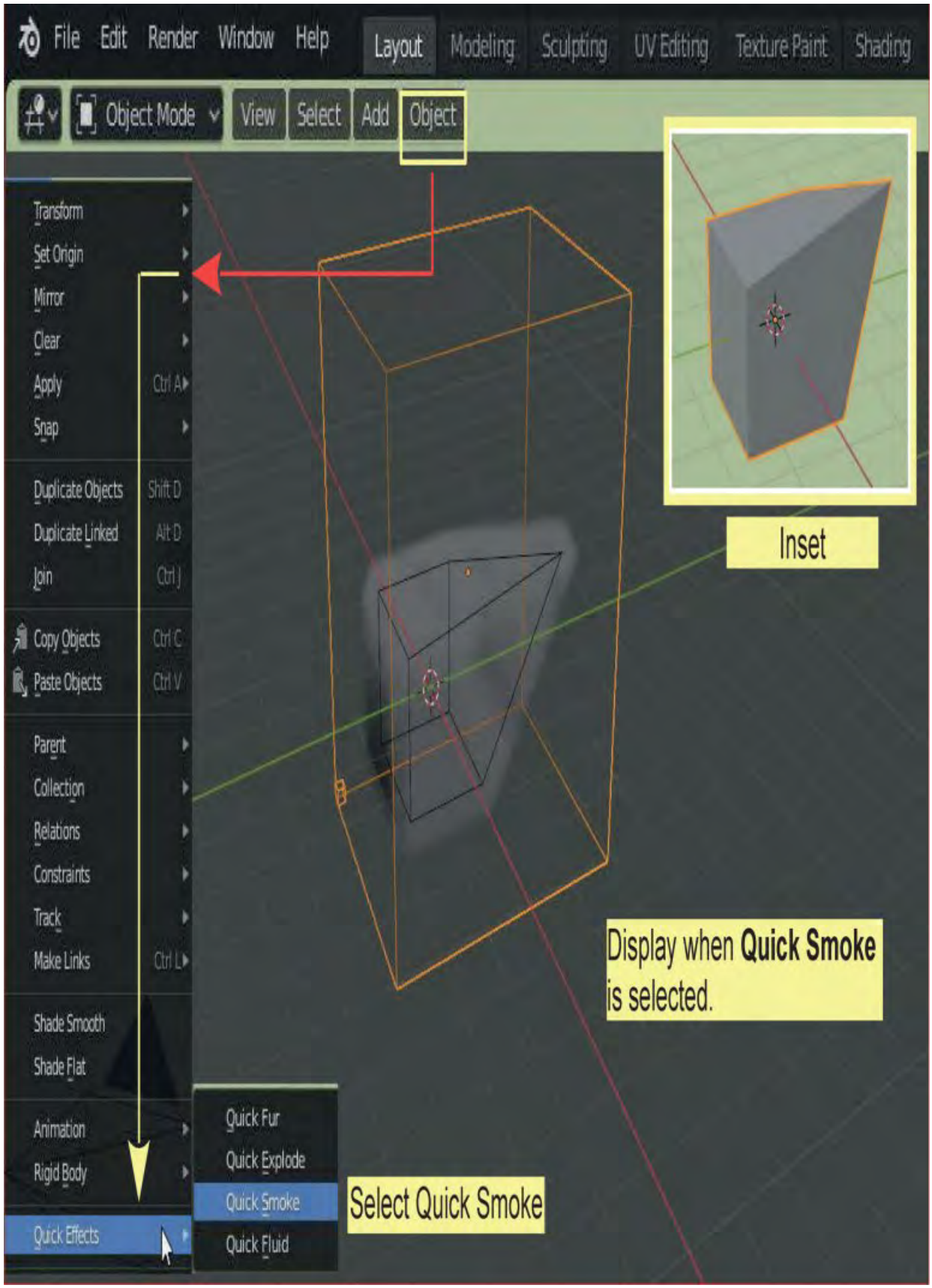

File
Edit
Render
Window
Help
Layout
Modeling
Sculpting
UV Editing
Texture Paint
Shading
Object Mode
View
Select
Add
Object
Transform
Set Origin
Mirror
Clear
Apply
Ctrl A
Snap
Duplicate Objects
Shift D
Duplicate Linked
Alt D
Join
Ctrl J
Copy Objects
Ctrl C
Paste Objects
Ctrl V
Parent
Collection
Relations
Constraints
Track
Make Links
Ctrl L
Shade Smooth
Shade Flat
Animation
Rigid Body
Quick Effects
Quick Fur
Quick Explode
Quick Smoke
Quick Fluid
Inset
Display when Quick Smoke is selected.
Select Quick Smoke

Figure 1.29

When you click on **Quick Smoke** the display of the modified Cube changes to that shown in Figure 1.29. **DO NOT CLICK in the 3D Viewport Editor**. **Note** the **Quick Smoke button** in the lower left hand corner of the 3D Viewport Editor (Figure 1.30). The inset in the upper right hand corner of Figure 1.29 show the Cube before clicking Quick Smoke.

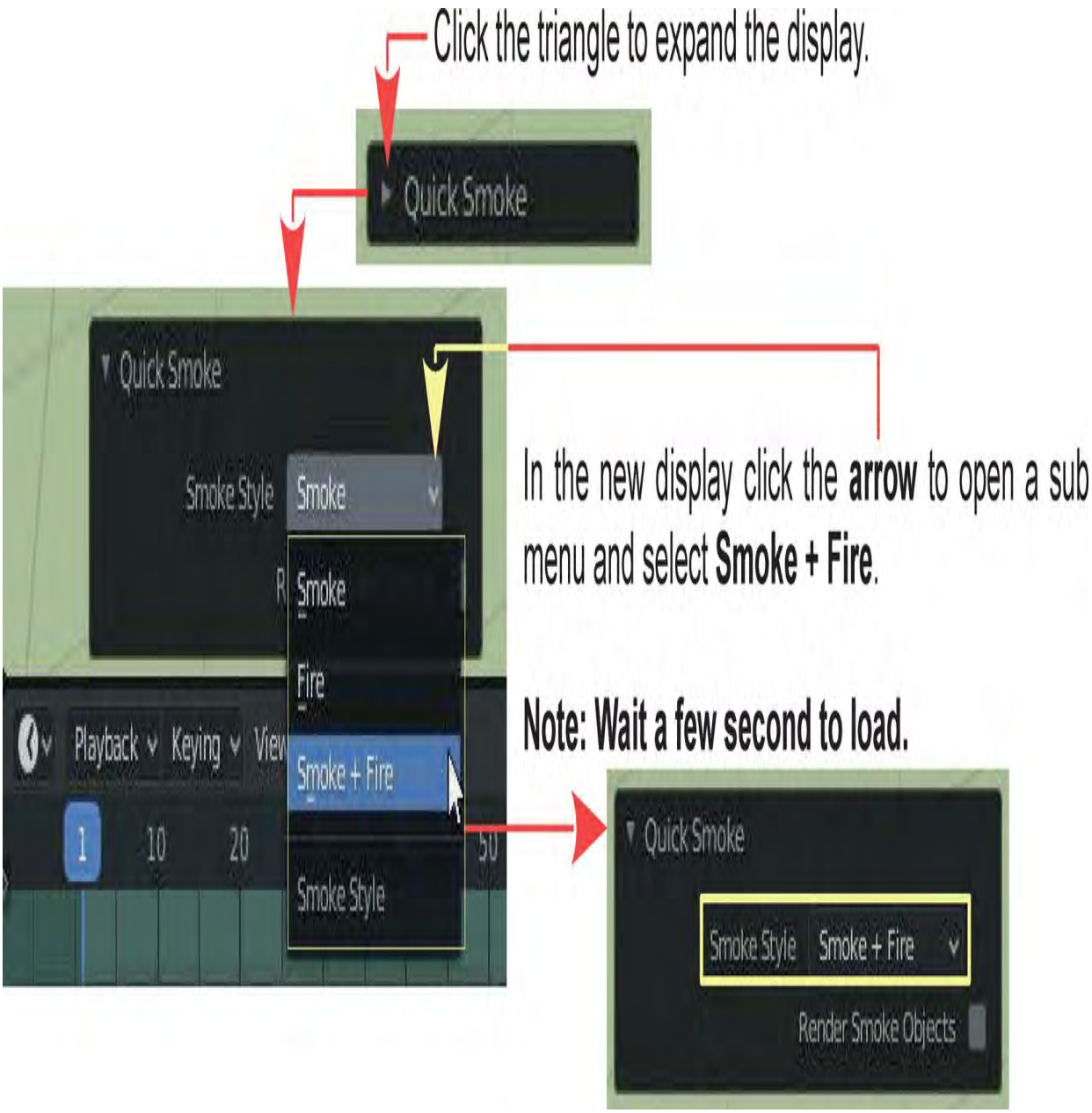


Figure 1.30
Figure 1.31

The foregoing set up a **Smoke + Fire Simulation** (see Chapter 25 – 25.11). In the **Timeline Editor** (Figure 1.32), click the **Play** button to see smoke and fire billowing from the Cube (Figure 1.33, **Note:** The 3D Viewport Editor in the default **Solid Viewport Shading Mode** (see Chapter 14).

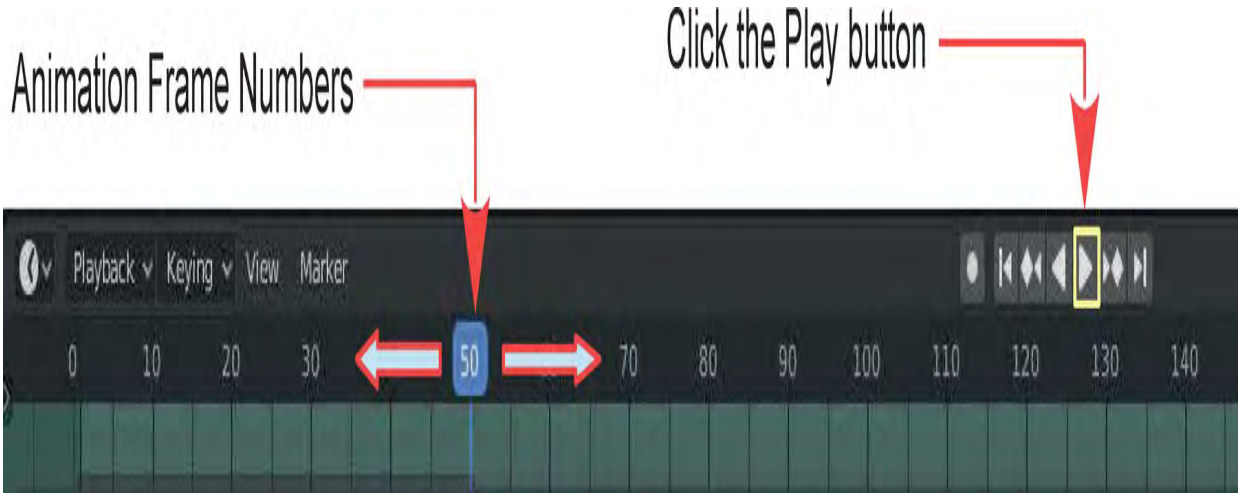
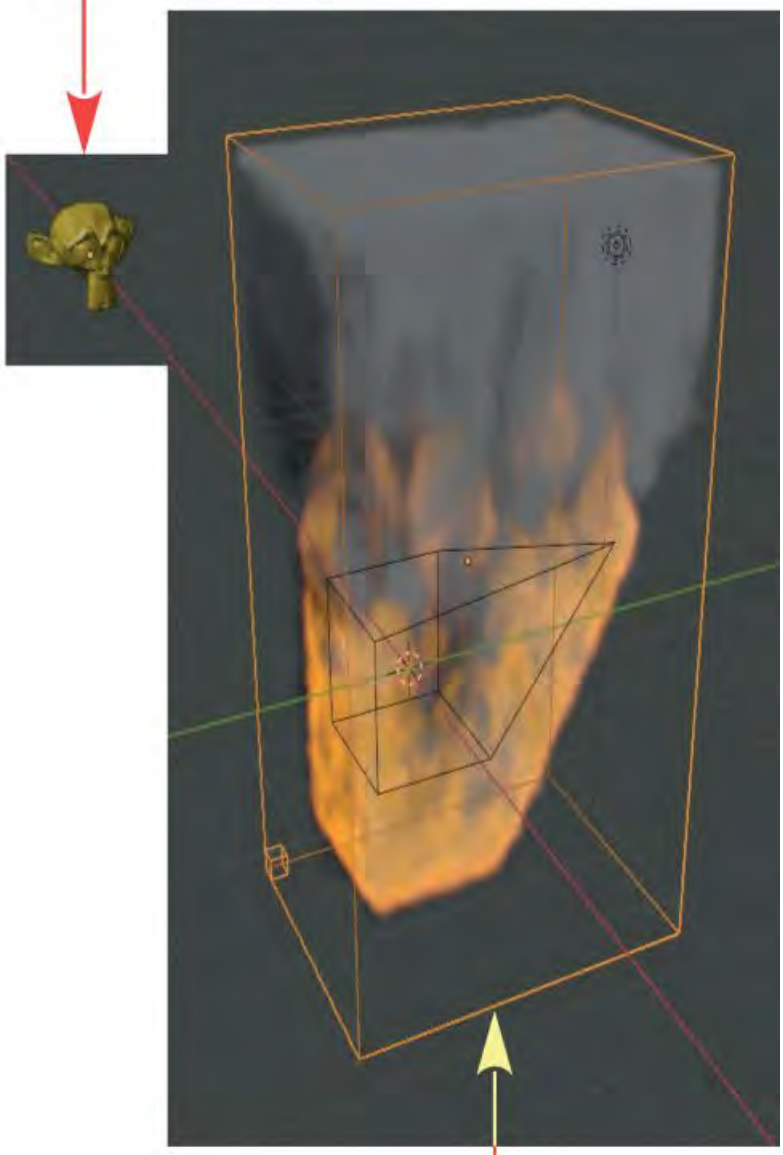


Figure 1.32

The Animation stopped at Frame Number 50

Suzanne moved back in the Scene



By scaling the **Cube** down, reshaping the volume in which the simulation takes place and adding a Force Field the smoke and fire can be directed to suit the action intended in a Scene (see Chapter 25 – 25.11).

Figure 1.33

Clicking the **Play** button in the **Timeline Editor** generates an Animation Sequence. When generating the animation is very slow and jerky.

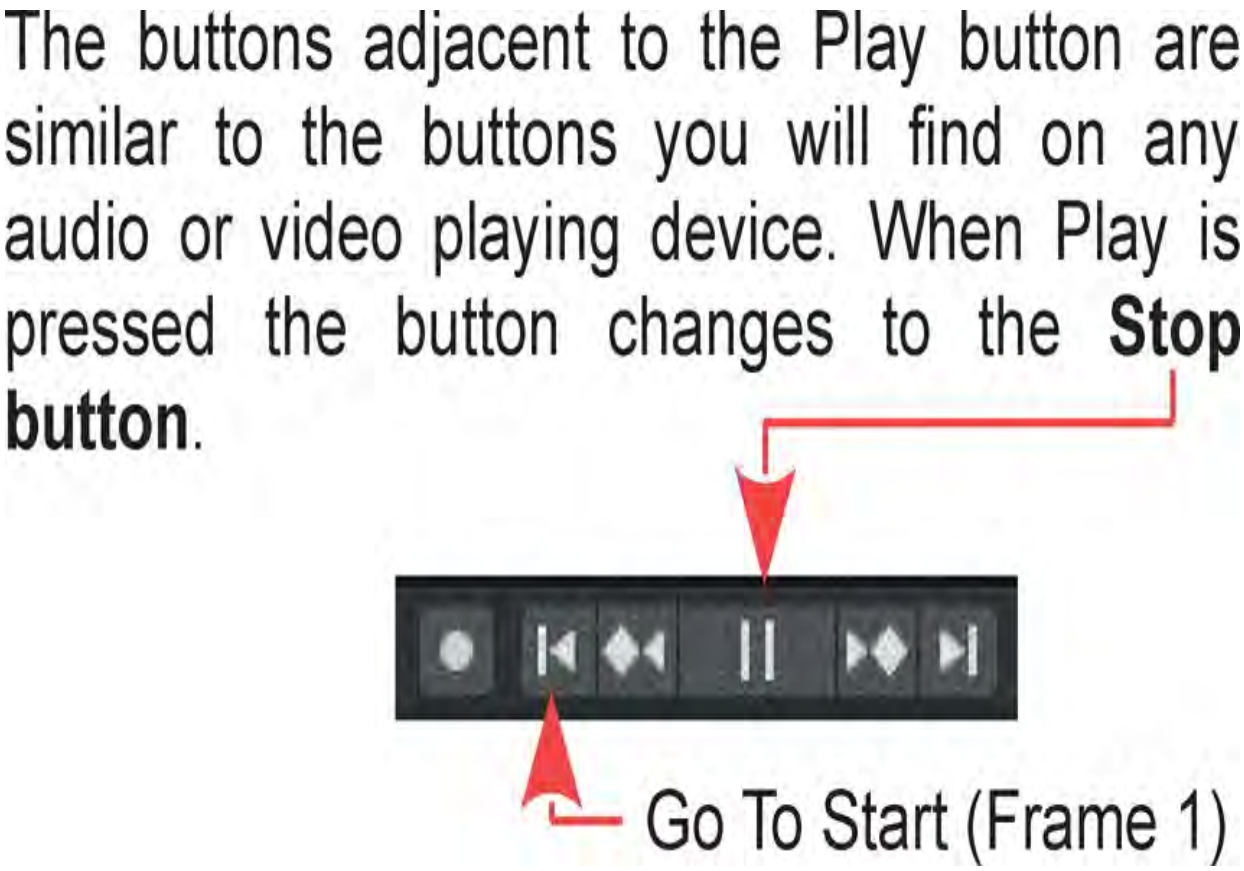


Figure 1.34

The default animation length is 250 Frames as seen by the **End Frame** number below the Header. When the Animation reaches Frame 250 it automatically replays. The replay is much smoother.

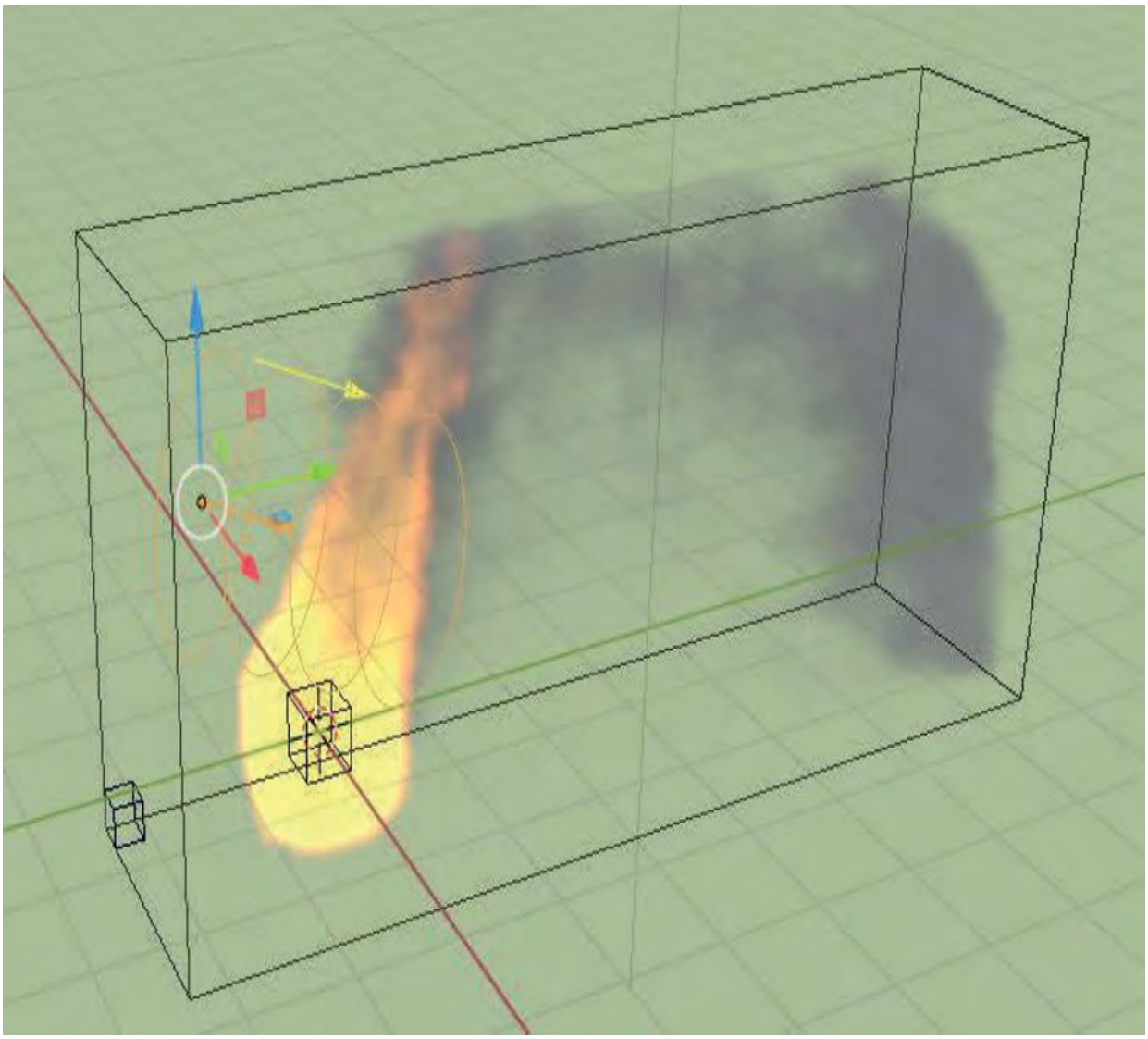


Figure 1.35

With the fire raging Suzanne's face is still blue with cold. It would be nice to invite her to come close to the warmth.

1.14 Animation

There are many properties which can be Animated. To give you an example, Suzanne will move in towards the fire and warm up (change color).

Select Suzanne in the 3D Viewport Editor. In the Timeline Editor, have the Timeline Cursor (blue line) at Frame 1 (Press the Go to Start button). The

smoke disappears.

With Suzanne selected and the Mouse Cursor in the **3D Viewport Editor**, press the **I Key**. In the menu that displays select **Location**.

In the Timeline Editor click on Frame 60 moving the Timeline Cursor to Frame 60 (smoke reappears).

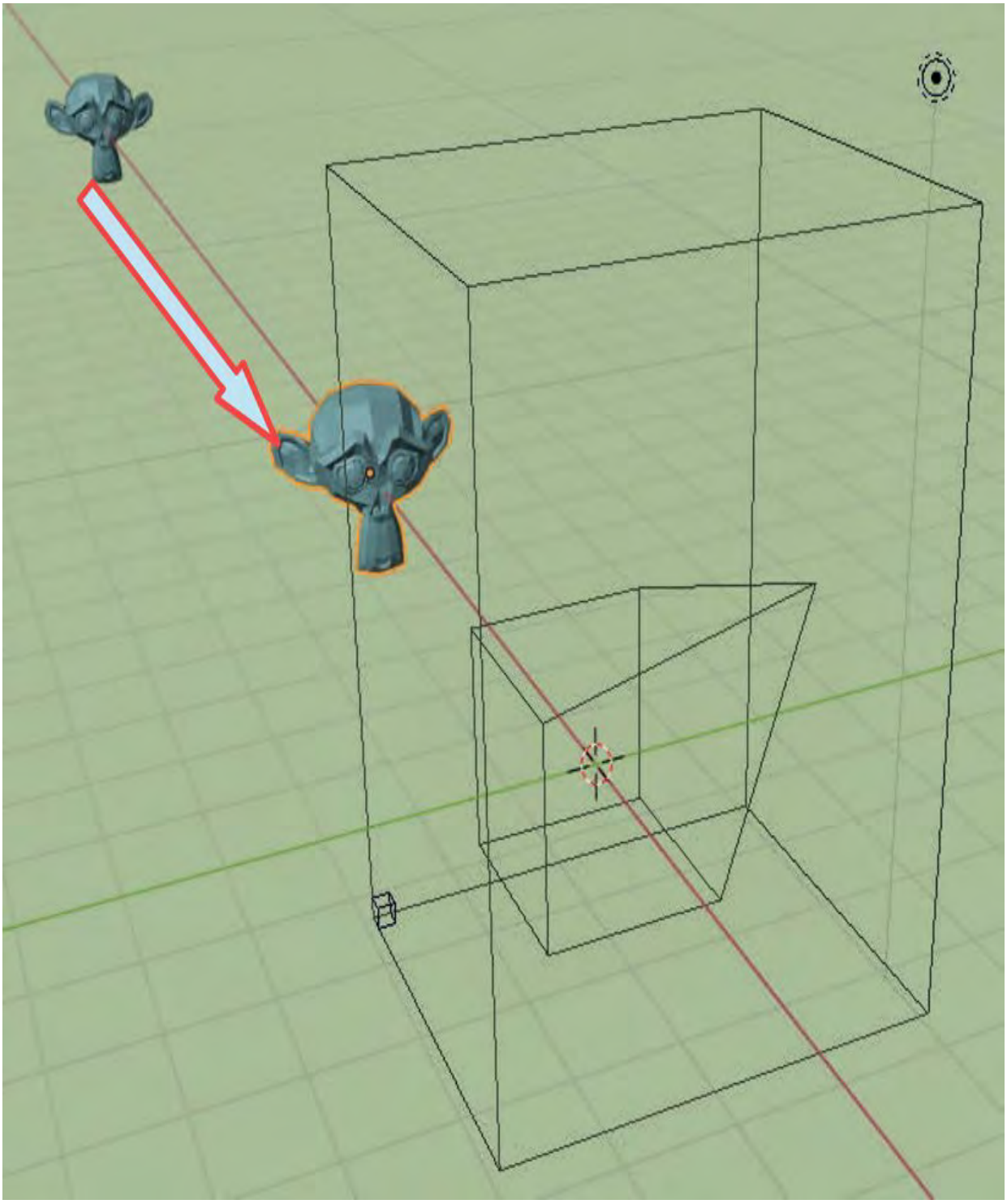


Figure 1.36

Press the **G Key + X Key** and drag the Mouse moving Susanne in towards the Cube along the X Axis. With the Mouse Cursor in the 3D Viewport Editor,

press the **I Key** for the Insert Keyframe menu and select **Location**.

When the Animation is replayed the fire starts and Suzanne moves close to the the heat but she doesn't appear to warm up. She still looks pretty blue. Return the Animation to Frame 1 and head over to the **Properties Editor**.

Have **Suzanne selected** at Frame 1 in the Animation Sequence. Have the **Material Properties** selected in the Properties Editor **with Use Nodes cancelled**. **Right Click** on the blue Base Color bar and click on **Insert Keyframe** in the menu that displays.

Click on Frame 60 in the Timeline Editor. Click **LMB** on the blue Base Color bar and select a nice rosy red hue in the color picker.

Click **RMB** on the rosy red **Base Color bar** and select **Insert Keyframe**.

Figure 1.37

Click **RMB** on the rosy red **Base Color bar** and select **Insert Keyframe**.

Playing the Animation shows the Fire burning with Suzanne moving in towards the warmth changing from cold blue to warm red.

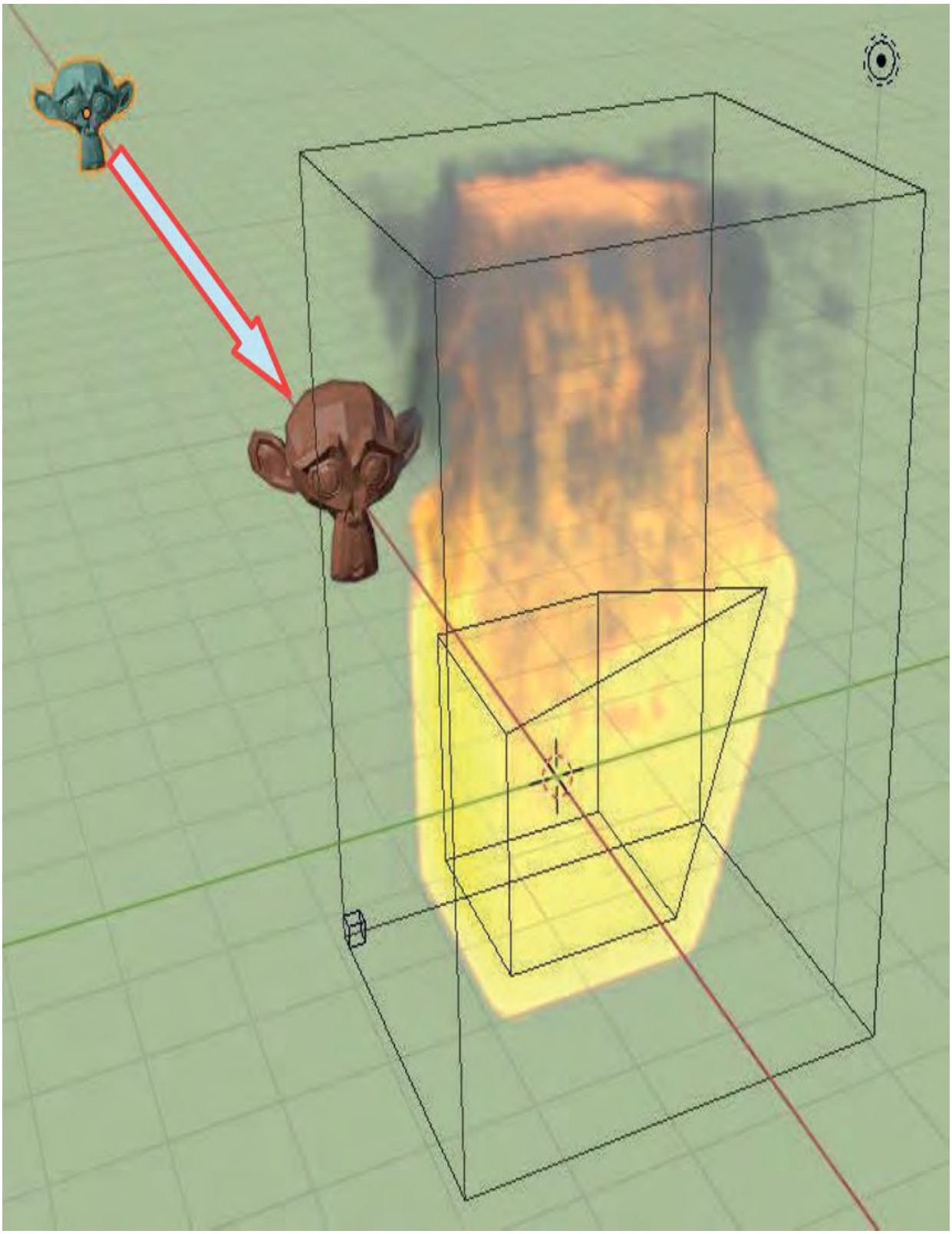


Figure 1.38

This has demonstrated only two properties which may be animated. A more detailed discussion is found in Chapter 20.

1.15 Summary

The forgoing is intended to familiarise you with the Graphical User Interface and demonstrate a few of the operational features of Blender. Understanding the Interface and how the Editors relate to one and other is a start to understanding how Blender works.

This has been a brief introduction. There are many features to be discovered, therefore the following chapters will expand on what has been covered and show where Blender Tools are located and give examples on how to use the Tools.

Before moving on to the other Chapters there are two more items you should be aware of.

Undoing: When you perform successive operations in creating a Scene you can backtrack through the sequence of operations, undoing each operation. By default undo operations are limited to 32 steps. To undo an operation press **Ctrl + Z Key**. This applies to any Editor.

Deleting: Deleting means, removing an Object and all its Properties from a Scene. To delete an Object, select the Object and press the **X Key**, click **Delete**. This applies only to the 3D Viewport

Editor.

Deleting: Deleting means, removing an Object and all its Properties from a Scene. To delete an Object, select the Object and press the **X Key**, click **Delete**. This applies only to the 3D Viewport Editor.

Figure 1.39

2 Editors – Workspaces - Themes

2.1 Editor Types
2.2 Resizing Editors
2.3 Splitting Editors
2.4 Cancel an Editor
2.5 3D Viewport Editor Features
2.6 Scene Manipulation
2.7 Scene Manipulation Widget
2.8 Multiple Scenes
2.9 Headers Menus and Panels
2.10 Headers and Panels
2.11 The Blender Interface Header
2.12 The Blender Screen Header
2.13 The 3D Viewport Editor Header
2.14 The Widget Panel
2.15 Tool Panel and Object Properties Panel
2.16 Properties Editor Tabs
2.17 The Preferences Editor
2.18 3D Viewport Editor - Background Color
2.19 Workspaces
2.20 Creating New Workspaces
2.21 Themes
2.22 Saving a Theme

Editors – Workspaces - Themes

Editors are the individual windows or panels which make up the Blender Interface. They contain the controls for editing data. Everything initially displayed in the 3D Viewport Editor (Window) is generated by a set of default data which you modify using the controls in the various Editors.

Workspaces are the arrangement of Editors in the **Graphical User Interface** (GUI). The Blender **GUI** opens with four separate Editors displayed. This arrangement constitutes a **Workspace** (working space) where you arrange models and characters and create Scenes. The default Workspace is called the **General** Workspace. There are alternative pre-constructed Workspaces available and you may configure your own.

Themes are how the Graphical User Interface displays as a whole and are purely cosmetic. Blender has several Themes for selection. You can create your own or download custom builds from the Internet.

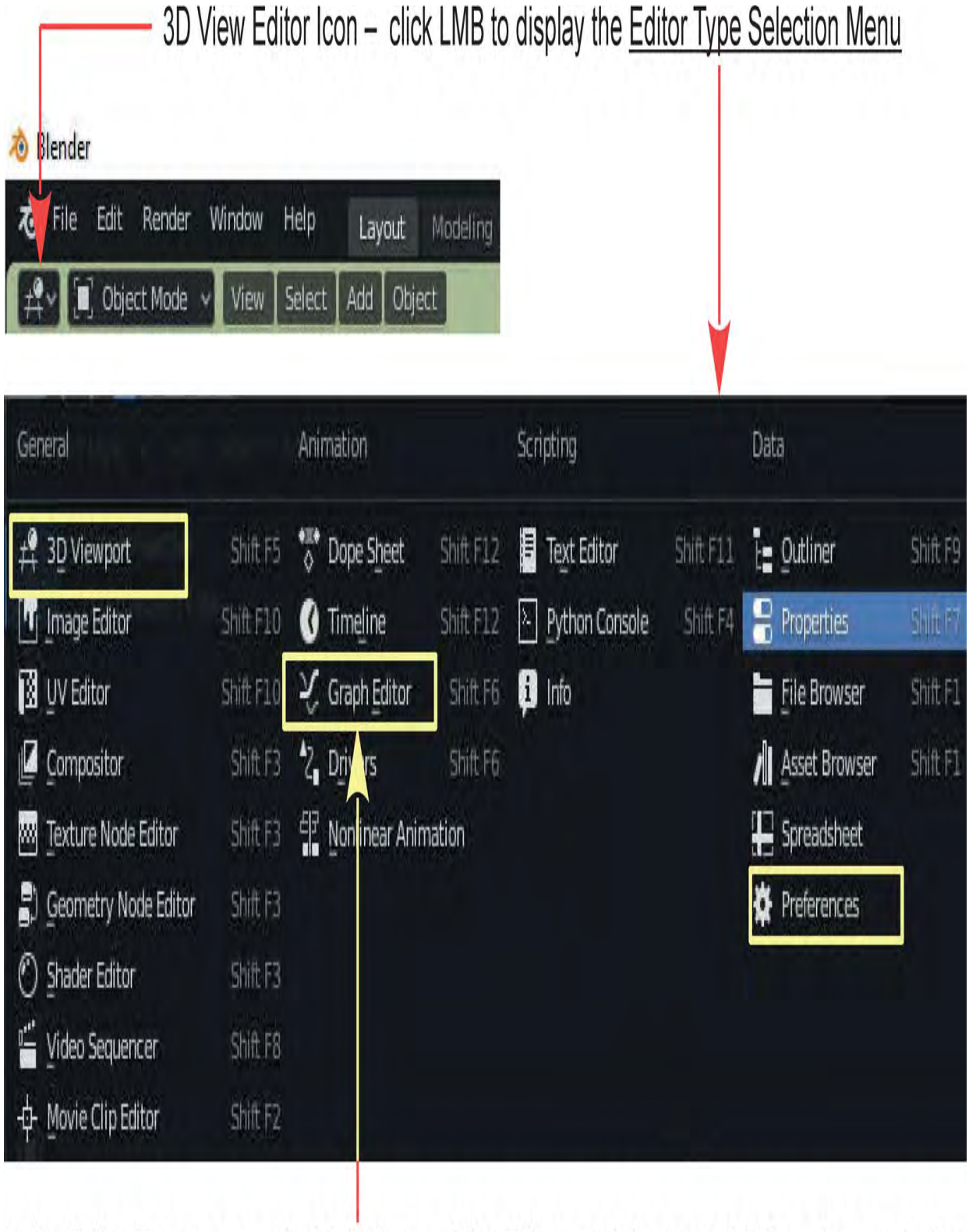
You may re size the Editor Panels, divide the Panels, creating additional Panels and change each Panel to a different type. In doing this you create a specialised Workspace which can be saved in a Blender file and reused. You can generate your personal Theme in a Blender file.

This chapter will introduce the controls for each of the above which will allow you to configure Blender for your personal requirements.

2.1 Editor Types

The **Default** Screen arrangement comprises four individual panels or windows, as shown in Figure 1.1 in Chapter 1. The panels are called **Editors**.

The default Editors displayed are: The **3D Viewport Editor**, the **Outliner Editor,** the **Properties Editor** and the **Timeline Editor**. Each has an icon representing the **Editor Type** in the upper left hand corner of the panel. Clicking LMB on this icon displays a menu for changing the Editor to a different Editor Type.



In the Editor Type menu, select (click) one of the Editors and the current Editor changes to that selected.

Figure 2.1

In the Editor Type menu, select (click) one of the Editors and the current Editor changes to that selected.

Here's an example; In the upper LH (Left hand) corner of the **3D Viewport Editor**, position the mouse cursor over the **Editor Icon** and click the left mouse button to display the Editor Type selection menu. Select (click on), **Graph Editor**, in the menu and the 3D Viewport Editor changes to the **Graph Editor**. In the upper left corner of this Editor click on the **Graph Editor** icon and select **3D Viewport** in the menu. The Graph Editor reverts to the 3D Viewport Editor. Any Editor may be changed to a different Editor type in this way.

Note: Make a note of the **Preferences Editor** in the selection menu.

2.2 Resizing Editors

Most Editors and panels may be resized. Place the mouse cursor on an Editor or panel border and it changes to a double headed arrow (Figure 2.2). Click and hold the LH Mouse button and drag the arrow to resize the Editor panel. This works on both horizontal and vertical borders.

Figure 2.2

2.3 Splitting Editors

Editors may be divided, to initially form a duplicate, then changed to a different Editor Type. When the mouse cursor is placed in the corner of an Editor panel it changes to a **white cross** Figure 2.3 Right). Click **RMB** and select a **Split Area** option in the menu that displays. Drag the Mouse **horizontally or vertically** into the Editor to be divided. The Mouse Cursor changes with a dividing line attached (Figure 2.3 Left). Position the dividing line and click LMB.

When the mouse cursor is placed in the corner of an Editor panel it changes to a **white cross** Figure 2.3 Right). Click **RMB** and select a **Split Area** option in the menu that displays. Drag the Mouse **horizontally or vertically** into the Editor to be divided. The Mouse Cursor changes with a dividing line attached (Figure 2.3 Left). Position the dividing line and click LMB.

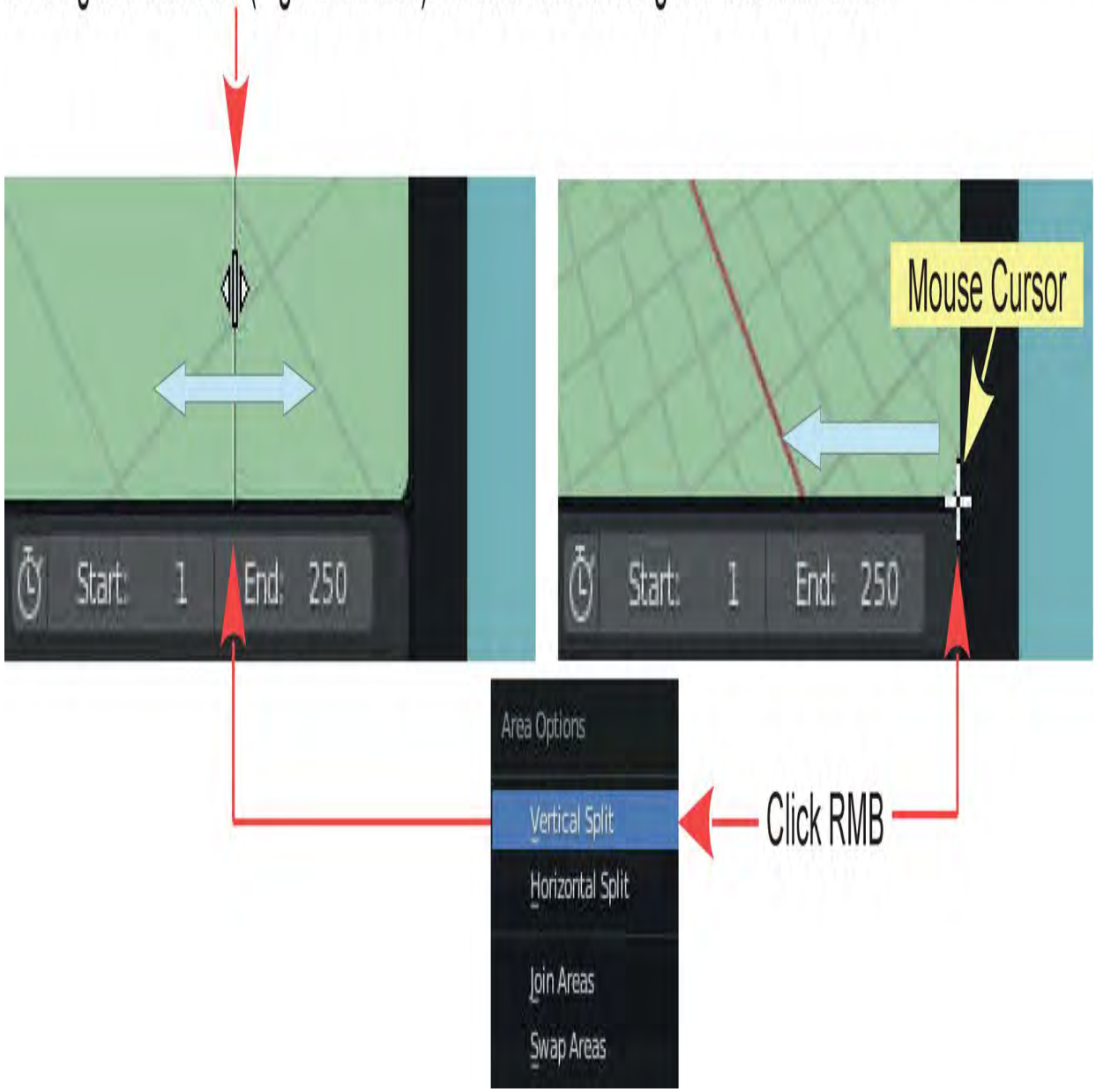


Figure 2.3

Alternatively, to split an Editor, mouse over in the corner of an Editor panel (mouse icon becomes a cross), LMB click, hold and drag the Cursor into the Editor to be divided.

If you make a mistake with the Split direction (Vertical – Horizontal) press **Esc** before releasing the Mouse button.

Alternatively, where **View** displays in a **Header**, click **View** then select **Area** (at the bottom of the menu that displays) then choose Horizontal or Vertical Split. Position the line that displays where you want the split to be then click LMB.

With an Editor split in two, one copy may be changed to another Editor Type (Figure 2.4).

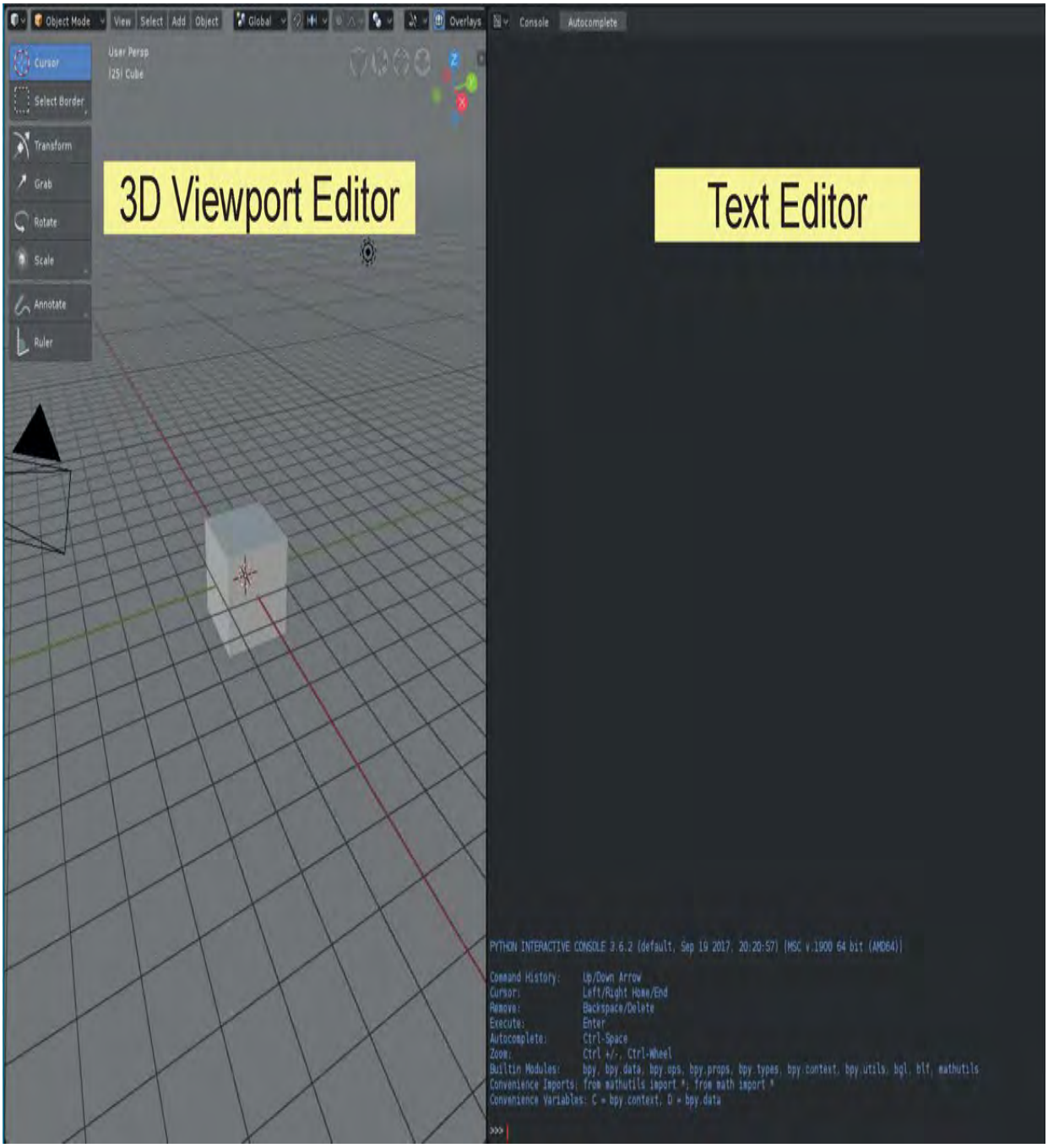


Figure 2.4

Note: Henceforth the **3D Viewport Editor** may be abbreviated to the **3D Viewport**

2.4 Cancel an Editor

To cancel or join an Editor panel, position the Mouse cursor in the Editor corner (cursor becomes a cross). Click RMB and select **Join Area** in the menu that displays (Figure 2.5). The Mouse Cursor becomes a chevron. Drag the chevron into unwanted Editor. The Editor darkens (Figure 2.6). Click LMB to cancel the unwanted Editor.

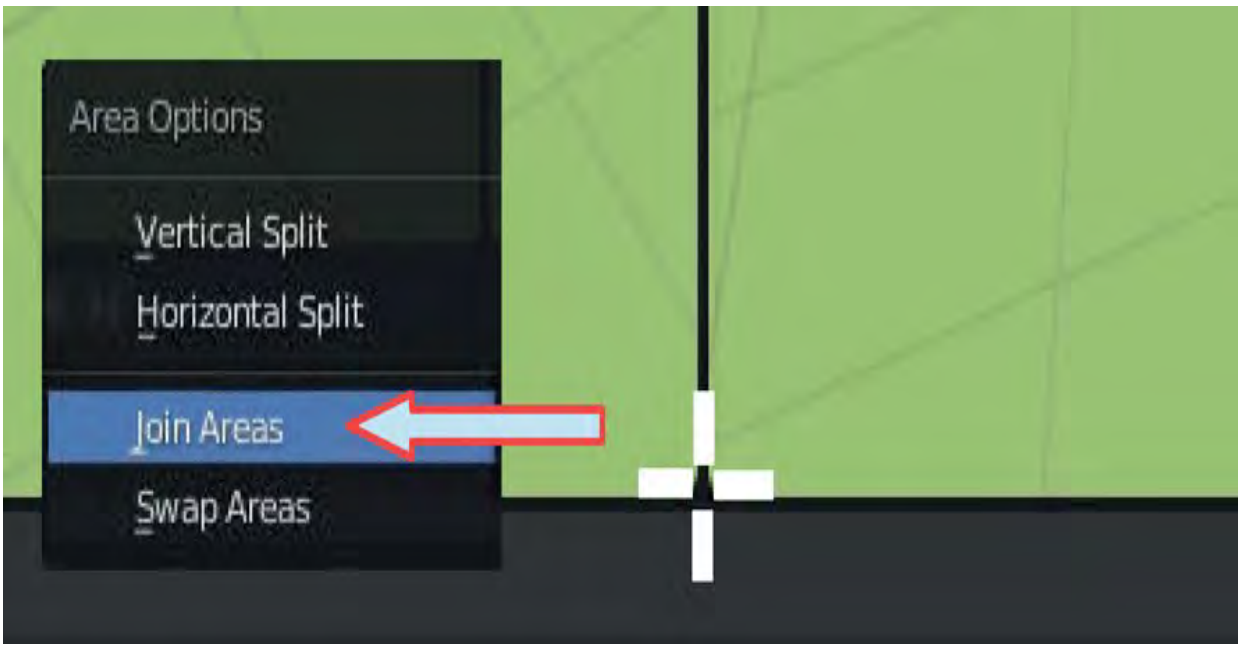


Figure 2.5

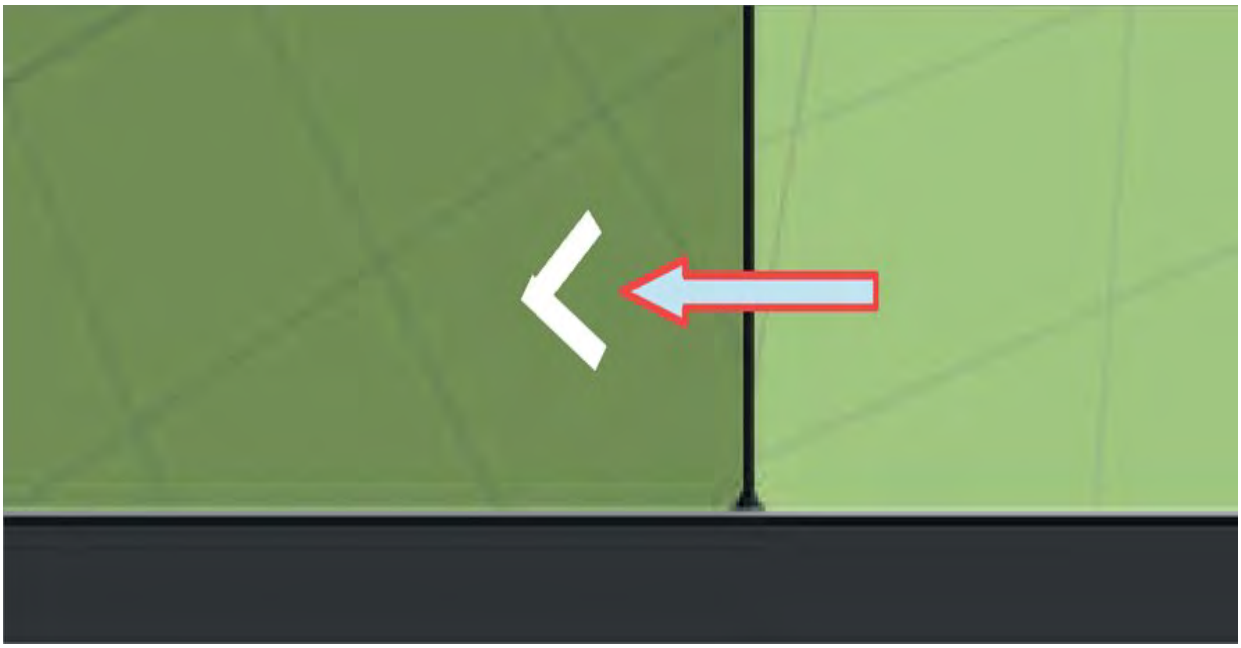


Figure 2.6

2.5 3D Viewport Features

Figure 2.7 shows the main features included in the **3D Viewport Editor** display.

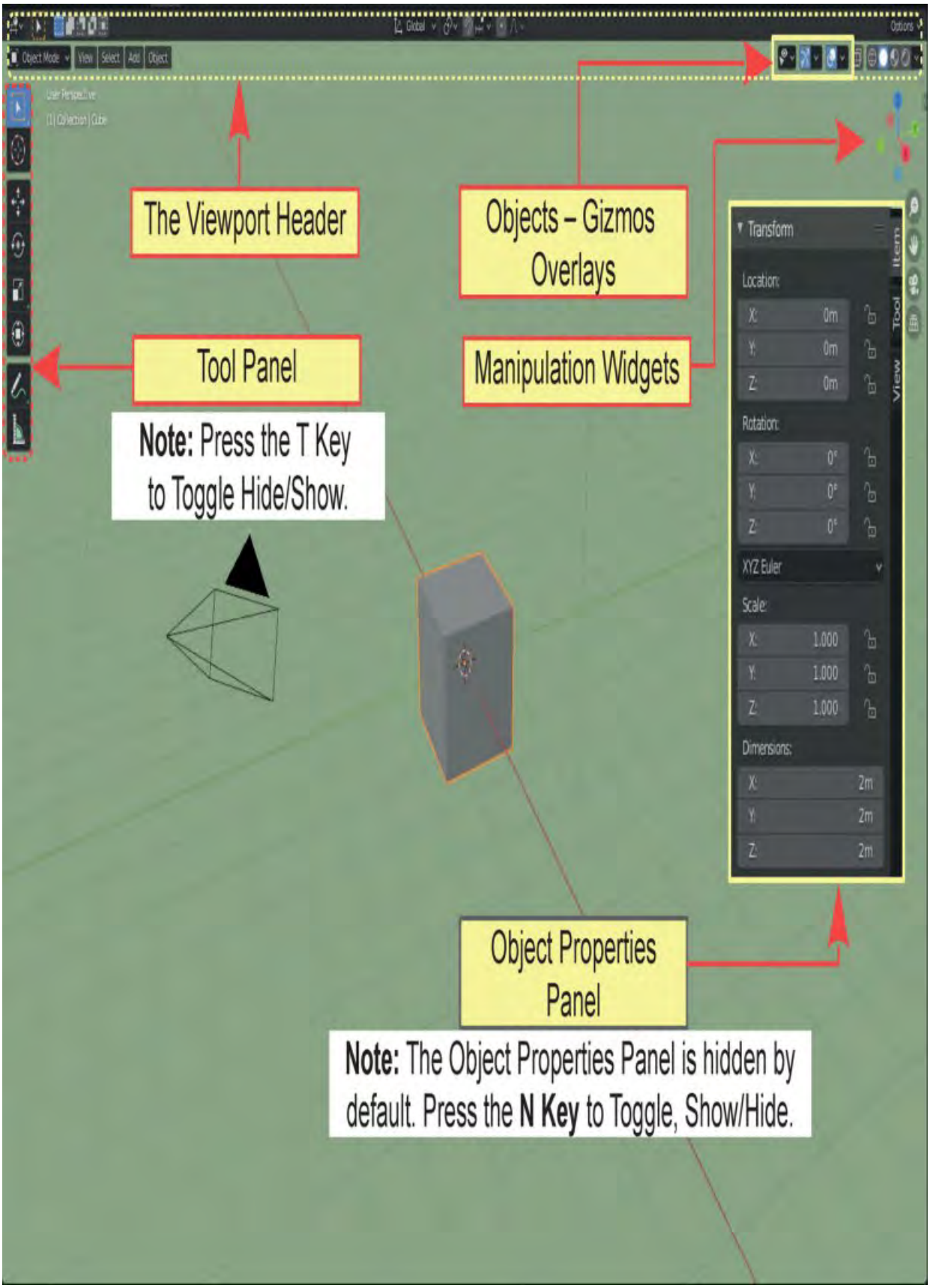

The Viewport Header
Objects – Gizmos Overlays
Tool Panel
Manipulation Widgets
Note: Press the T Key to Toggle Hide/Show.
Object Properties Panel
Note: The Object Properties Panel is hidden by default. Press the N Key to Toggle, Show/Hide.
Transform
Location:
Rotation:
XYZ Euler
Scale:
Dimensions:

Figure 2.7

Uncluttering the Scene

When a Scene becomes complicated it can be advantageous to turn off displays such as the User Perspective and Object notification in the upper left hand corner of the 3D Viewport or the Scene manipulation Widget at the right hand side or perhaps the background grid in the 3D

Viewport panel.

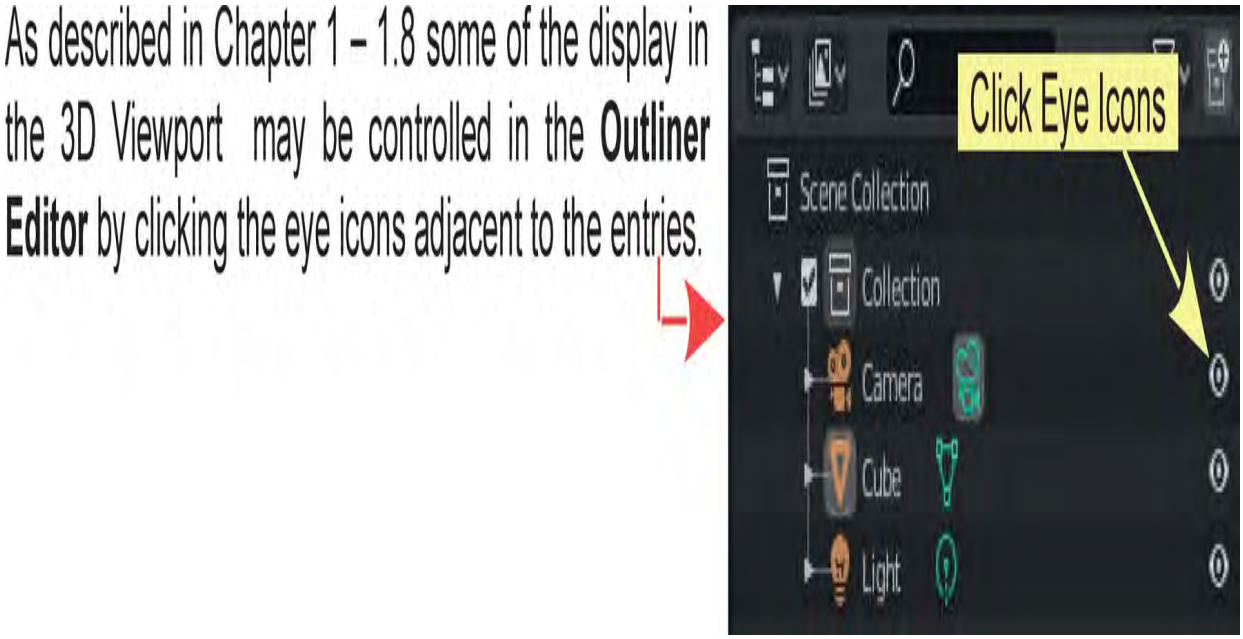


Figure 2.8

Hiding and showing displays here applies to Objects in the 3D Viewport.

Other displays are controlled by **Object Type**, **Gizmos** and **Overlays** from the 3D Viewport Header (Figure 2.9).

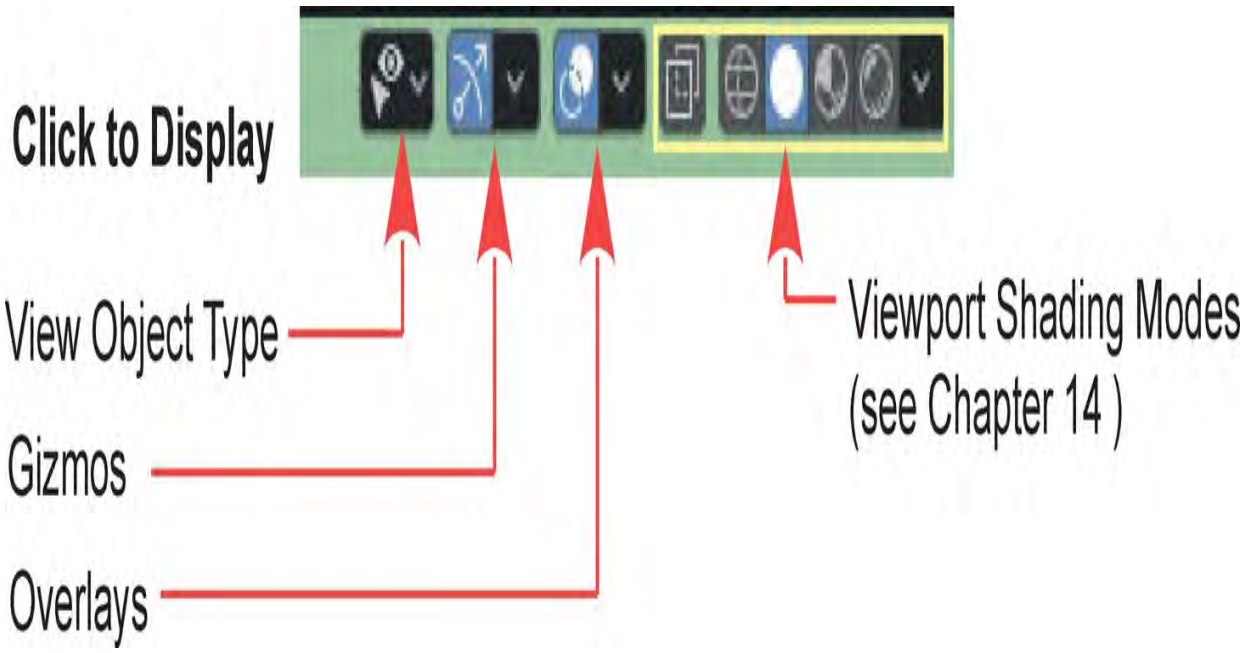
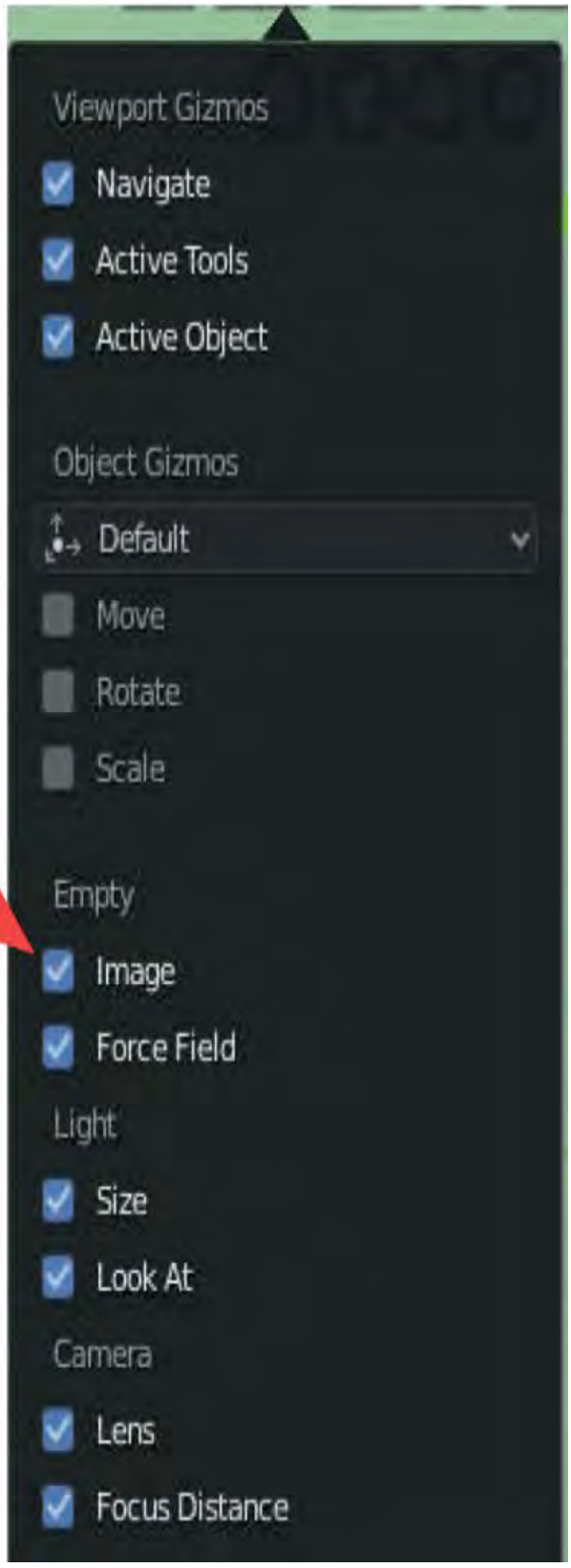

Click to Display
View Object Type
Gizmos
Overlays
Viewport Shading Modes
(see Chapter 14)

Figure 2.9

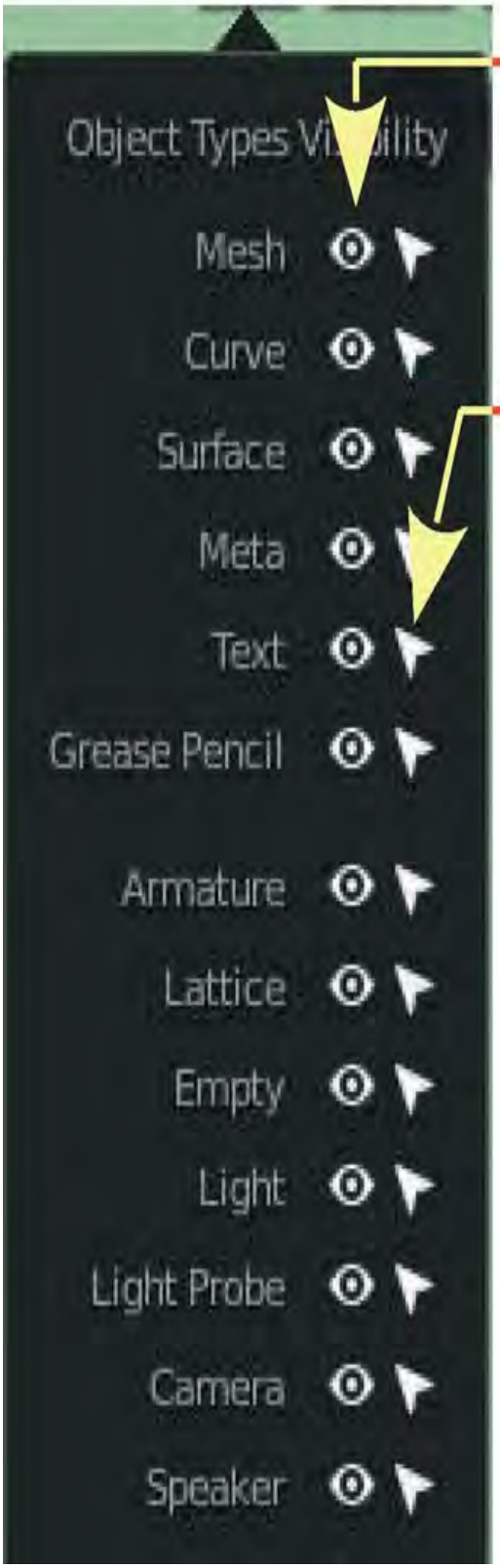
Viewport Gizmos



Check/Uncheck Gizmos to Activate/Deactivate functions.

Figure 2.10

View Object Type



Click an Eye Icon to Hide/Show an Object Type.

Click a Cursor Icon to toggle Cursor Selection in the 3D Viewport Editor.

Figure 2.11

Overlays: control what displays in the 3D Viewport.

Note: The Overlay selection is different in Object Mode and Edit Mode (Figure 2.12 over).

Object Mode Overlays

Edit Mode Overlays

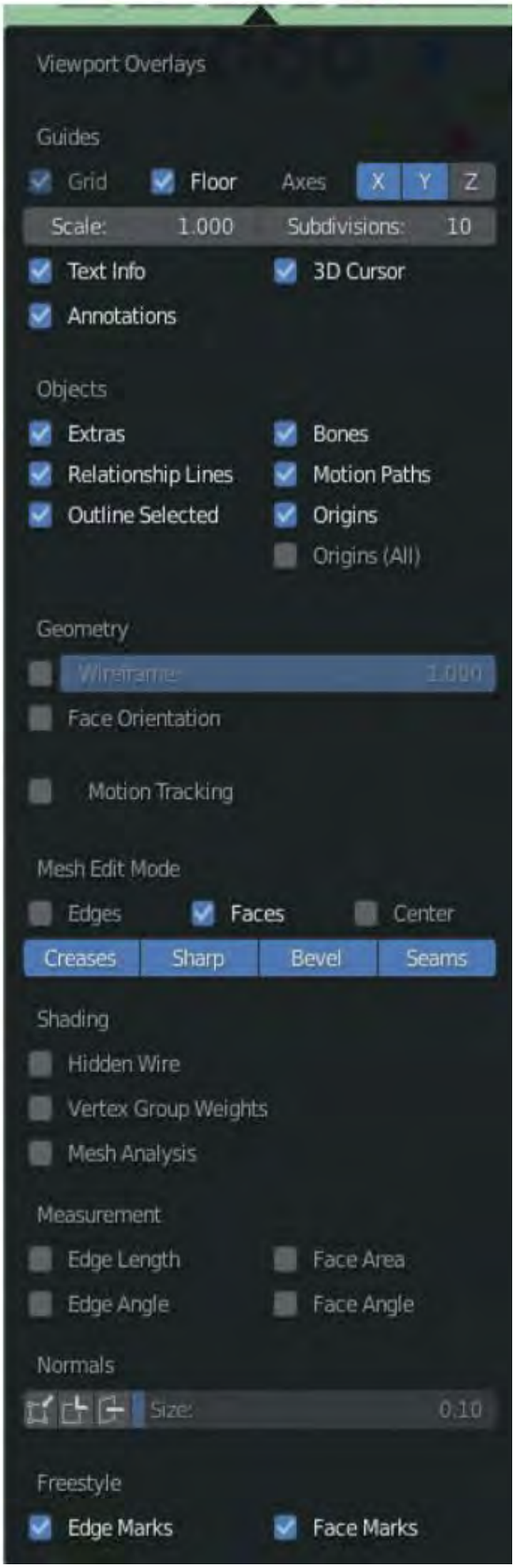


Figure 2.12

Note: The 3D Viewport has several different Modes. Overlays are only relevant to Object Mode and Edit Mode. You switch between the Modes by clicking on the Object Mode button in the 3D Viewport Header and selecting Edit Mode or vice verse. You may also toggle between Object and Edit Modes by pressing the **Tab Key**.

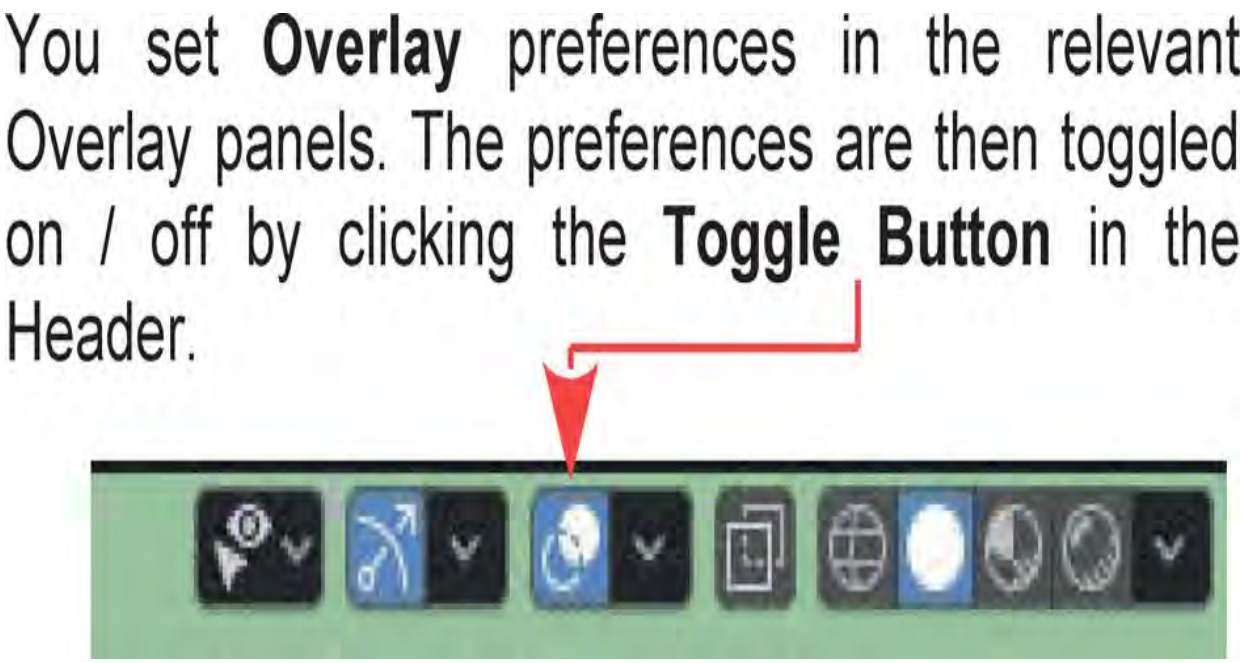


Figure 2.13

e.g. With **Floor** checked in Object Mode, clicking the Toggle button Hides and Shows the Grid Floor.

2.6 Scene Manipulation

Before adding new Objects and creating a Scene you should be conversant with how the 3D Viewport may be viewed and how to move in the three dimensional world.

Moving in 3D Space

In a 3D (Three Dimensional) program, not only do you have to consider where you are in two dimensions (height and width), but you also need to consider depth (how close or far away).

Moving around in the 3D Viewport is controlled by the Mouse and the Keyboard Number Pad.

User Perspective and Orthographic View

The Blender default Scene in the 3D Viewport, opens in the **User Perspective** view as indicated in the upper LH corner of the Editor. The Scene contains a Cube Object located at the center. There is also a Camera and a Light in the Scene. All three Objects are positioned relative to the center of the Scene which is the center of the 3D World, or if you like, a central point in 3D Space.

The Blender Scene may be viewed in either **Perspective** or **Orthographic** view.

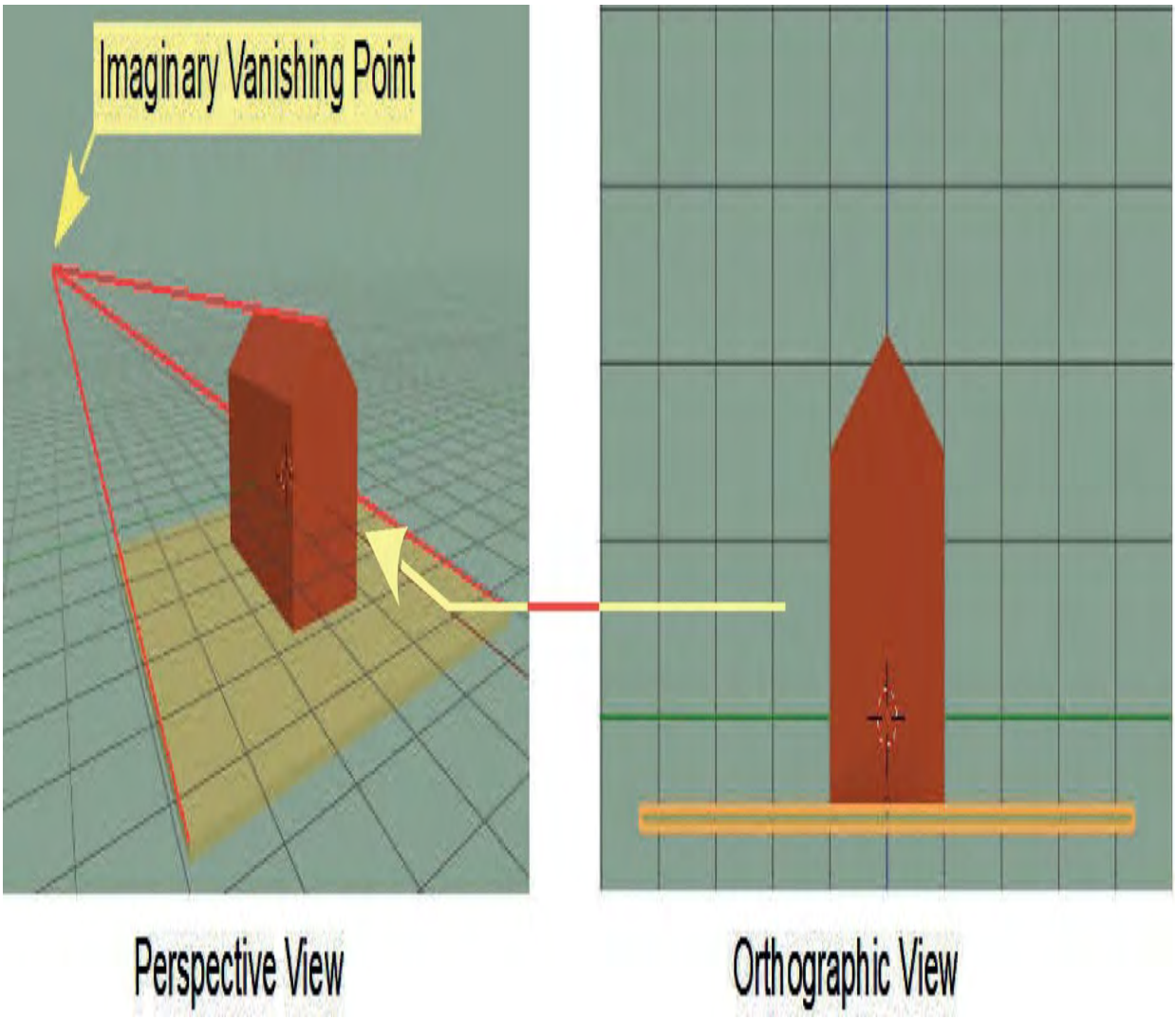


Figure 2.14
Figure 2.15

A **Perspective View** projects parallel lines to a single vanishing point somewhere in the distance. In landscape drawing this will be on the horizon.

An **Orthographic View** is seen looking square on (90º) to a Face.

The position of Objects relative to each other is important when considering 3D Space especially with Lights (lighting) and the Camera (seeing). When taking a photograph with a camera the position of the camera relative to what you want to photograph and where the lighting is located determine what you get in your snapshot. This is the same in a Blender Scene.

By default the Camera in the default Scene is positioned such that it points towards the Cube and with the default settings, for the Camera, captures an image of the Cube in its viewport. This is the image that will render (convert what the Camera sees to an image). To understand this, perform the following demonstration.

Render Demonstration

With the default Scene, have the Mouse Cursor in the 3D Viewport and press the Keyboard **Num Pad 0.** This places the 3D Viewport in **Camera View** (what the Camera sees). Press **F12** on the

Keyboard to **Render (Esc to Quit)**.

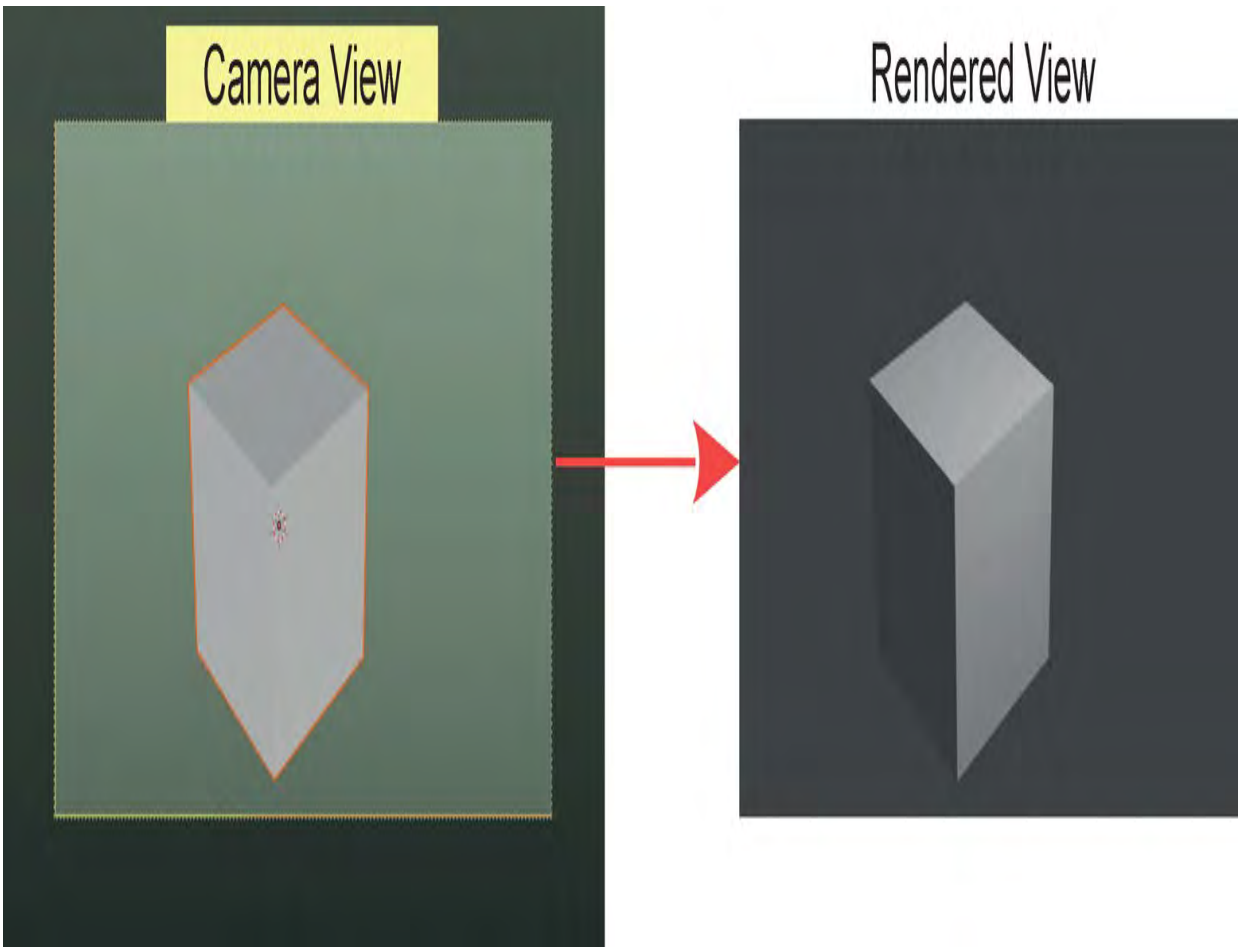


Figure 2.16

Pressing **F12** Renders (converts) the Camera view to a format that may be saved as an image file. Note that the rendered view has been opened in a new Editor – the **Image Editor**.

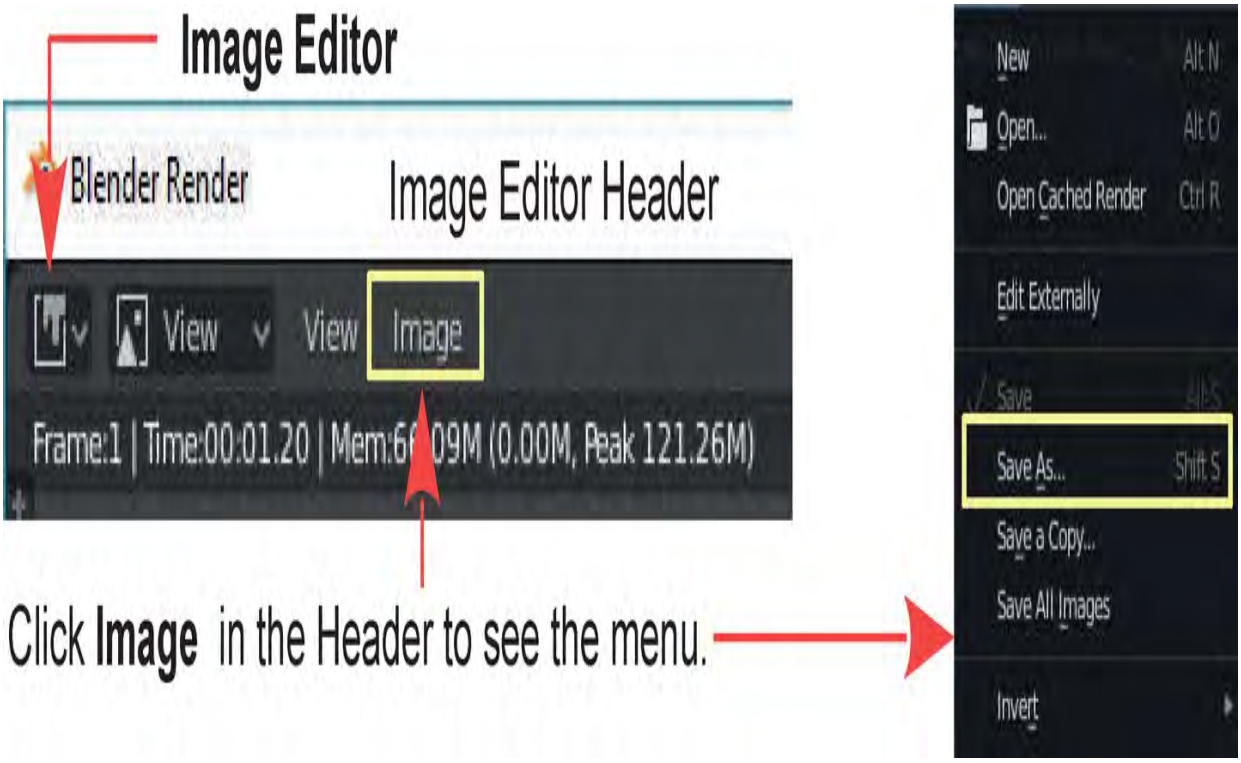


Figure 2.17

At this point you could save the display as an image. To save an image you have to understand how to save a file into a folder (see Chapter 3).

Press the **Esc Key** to cancel the Image Editor window and return to Camera View in the 3D Viewport . Press the **Num Pad 5 Key** to enter **User Orthographic** view, then press the **Num Pad 5 Key** a second time for **User Perspective** view.

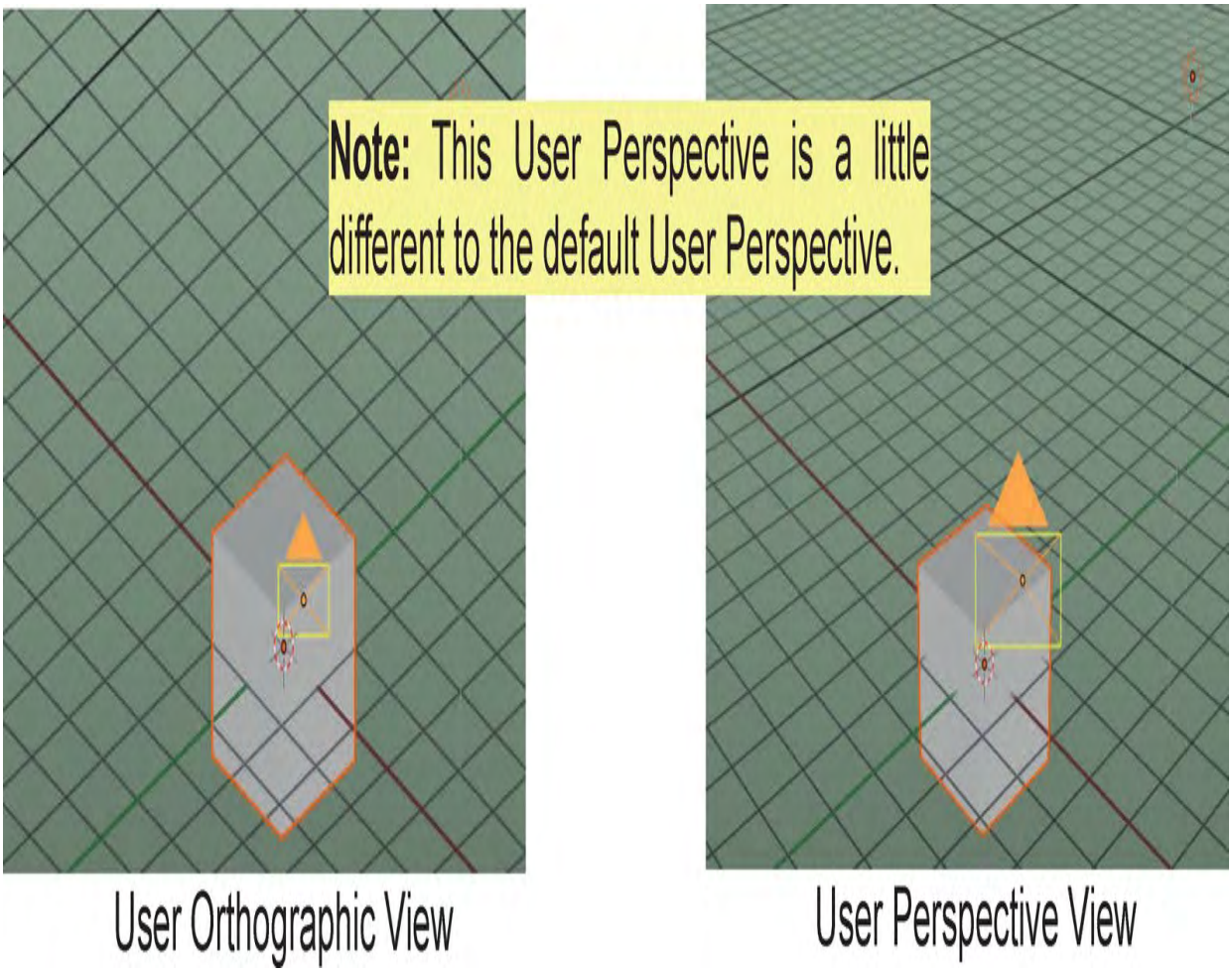


Figure 2.18

Pressing Num Pad 5 a second time displays a User Perspective view but it isn't the same as the original view in the 3D Viewport. To return to the original view press **Num Pad 6** or place the Mouse Cursor in the 3D Viewport, click the Middle Mouse Button (MMB), hold the button depressed and drag the Mouse to rotate the view.

Experiment with the options to find out what each one does. Remember you can always click on File – New – General in the Blender Screen Header to start over.

Clicking, holding and dragging MMB is one of several methods for manipulating the Scene in the 3D Viewport. Clicking **View – Navigation** in the Header gives a menu with all the **options**.

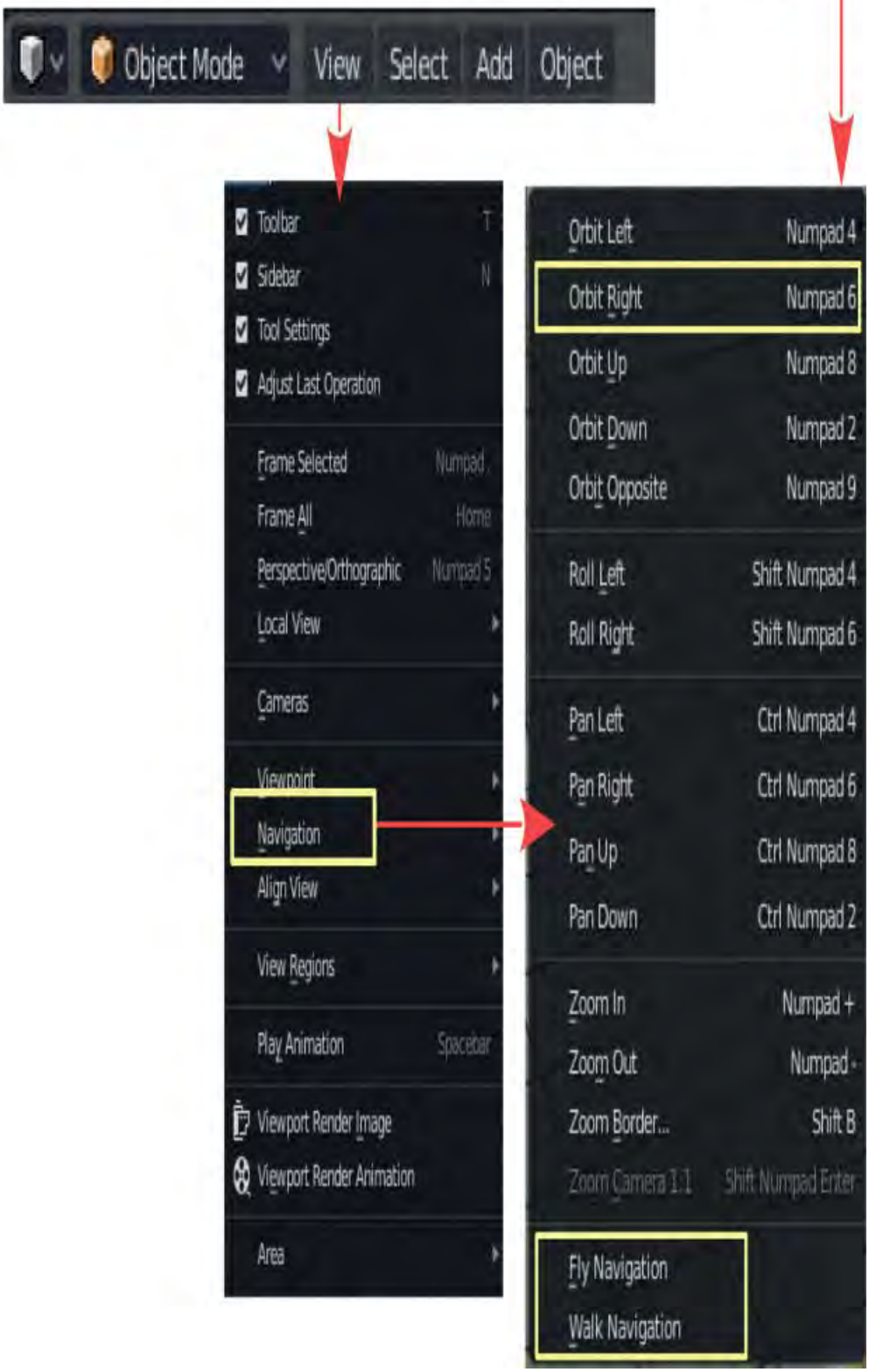


Figure 2.19

Two interesting navigation modes are **Fly Navigation** and **Walk Navigation.** After clicking Fly Navigation, moving the Mouse Cursor causes the view to FLY. **Move the Mouse gently**. Walk navigation is similar. In either case there are cross hairs in the center of the Viewport. Click LMB to cancel.

2.7 Scene Manipulation Widget (another way to Navigate)

Place the Mouse Cursor in proximity to the Widget to display the circle. Click, hold and move the Mouse to rotate the view.

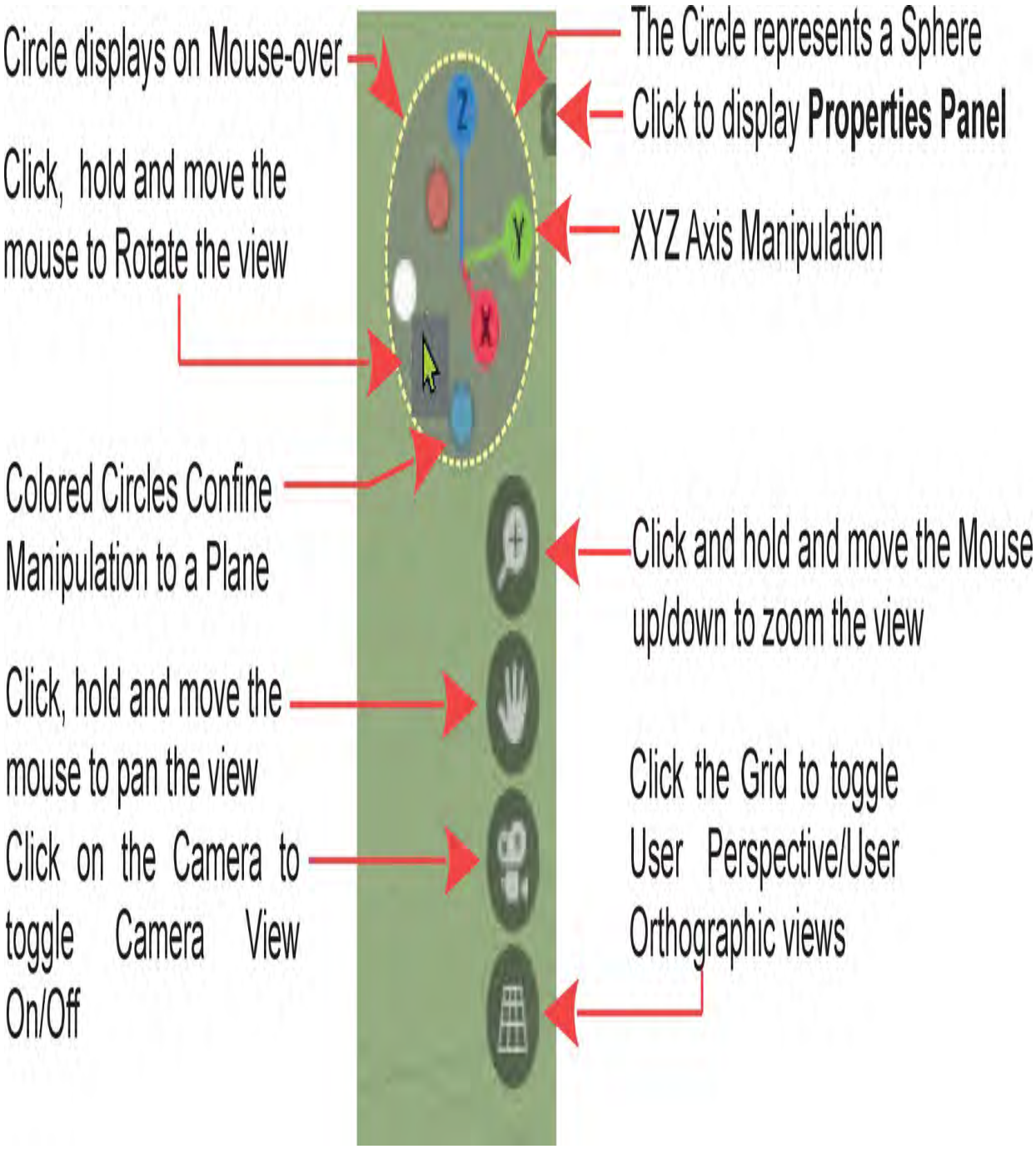


Figure 2.20

Note: The Red, Green and Blue circles on the rotation sphere align with the Red, Green and Blue Axis of the 3D Scene. Mouse over on a circle highlights the notation.

2.8 Multiple Scenes

At this point it is worth reinforcing the distinction between a **Scene** and a **Workspace**. The definitions have been presented at the beginning of the chapter but in introducing the concept of multiple Scenes clarification may be required.

The **Scene** is what you see in the 3D Viewport and consists of your models, characters, lighting effects and background. The **Workspace** is the arrangement of Editor Panels in the Graphical User Interface (what you see on the computer screen).

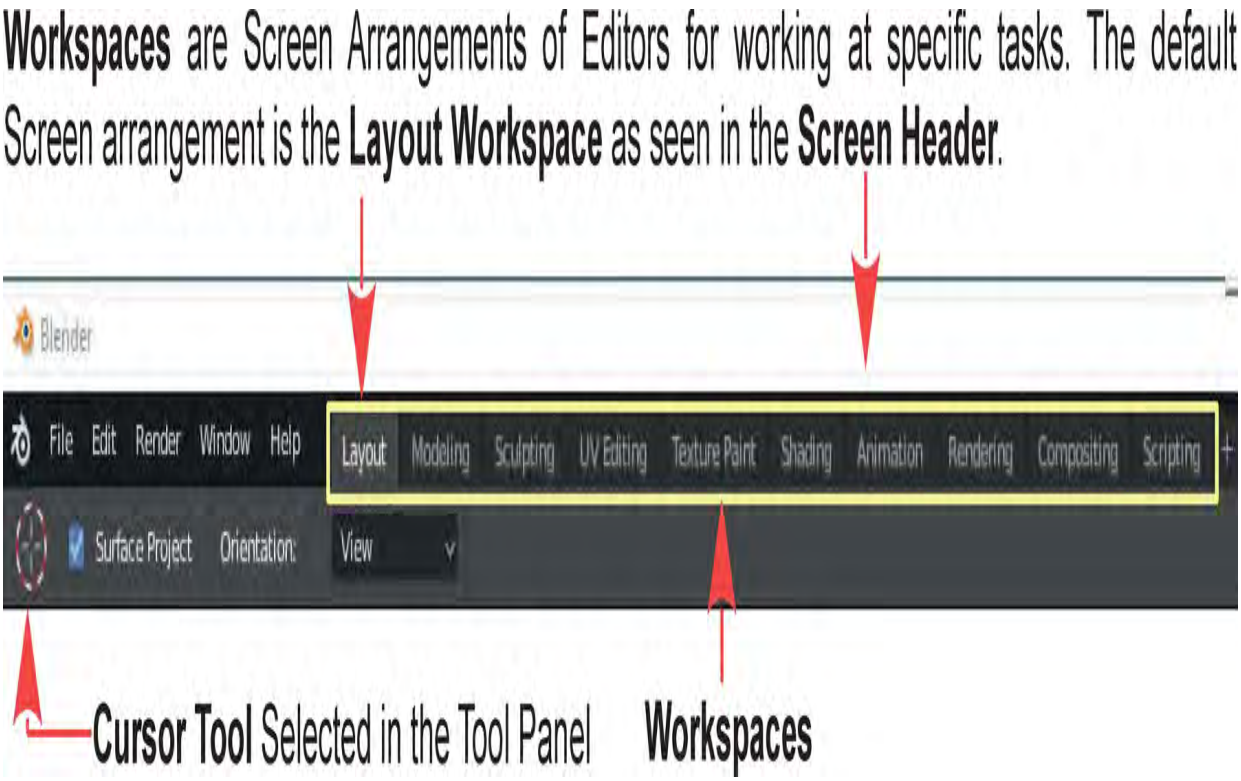


Figure 2.21

The main portion of the Layout Workspace is the 3D Viewport which shows a 3D View of the default Blender Scene. The Scene has a Cube at the center of the 3D World.

The Outliner Editor (Figure 2.22) displays **Scene Collection** which lists the Objects in the Scene in the 3D Viewport. At this point there is one single Scene in this Blender file.

Figure 2.22

At some stage you may wish to have more Scenes which is like having different sets on a stage or different sets for filming different parts of a movie.

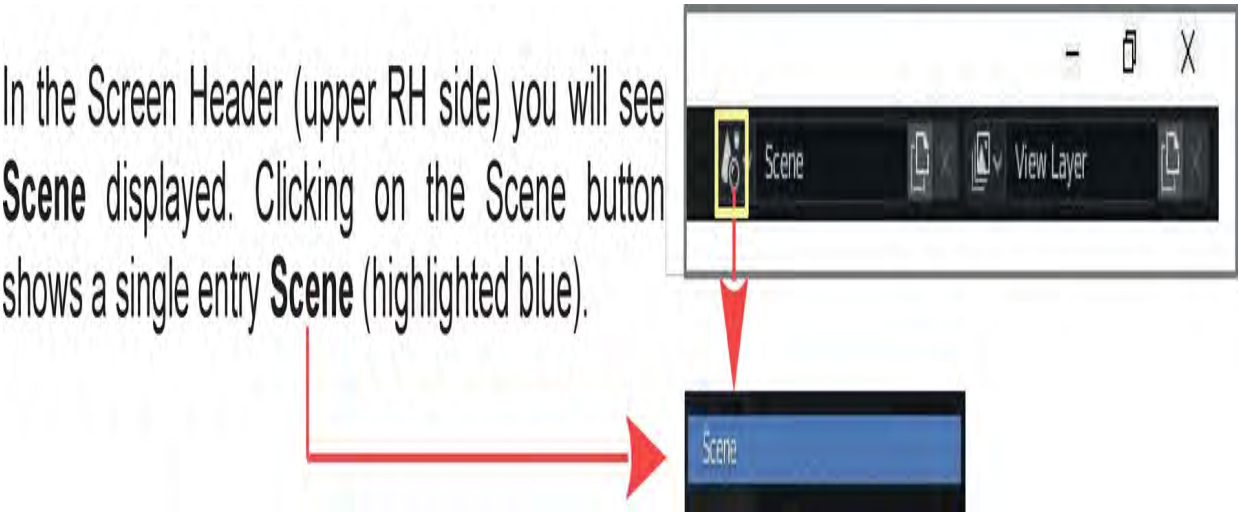


Figure 2.23

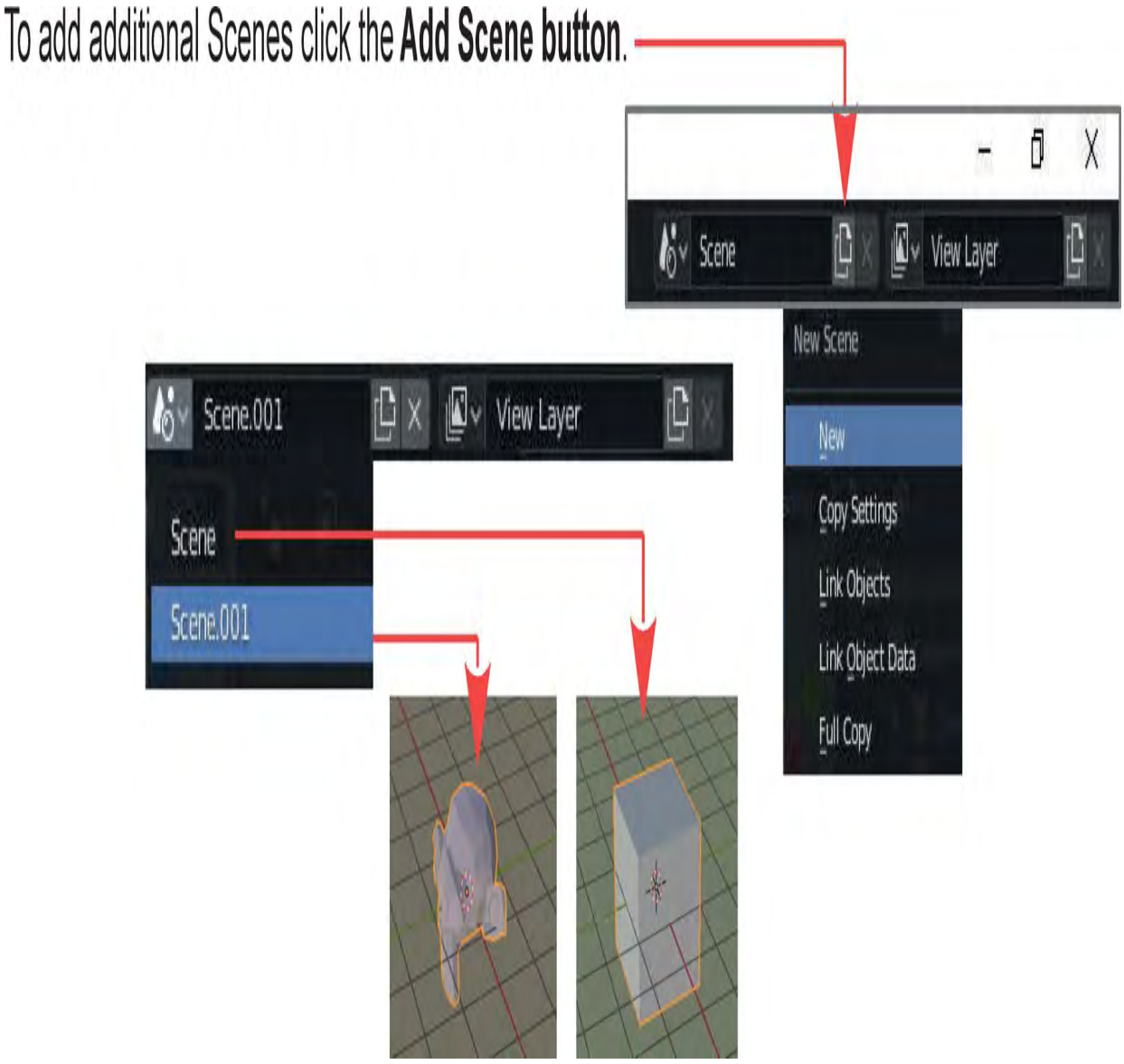


Figure 2.24

Click **New** in the options menu. The 3D View Editor shows a new empty Scene. **Scene.001** is entered in the Header.

If you add a Monkey to the Scene you have the option to select the default Scene with the Cube or the New Scene with the Monkey. The New Scene is

only available in the Blender File being worked on. Save the file for future use.

The demonstration shows the default Blender Scene containing the default Cube Object and a new Scene containing a Monkey Object. The default Scene is named: **Scene**, the new Scene is named: **Scene.001**.

With the two Scenes created make note of the content of the Outliner Editor when each Scene is selected in turn (Figure 2.25).

Click to display Scene selection

Figure 2.25

With **Scene** (the default Scene) selected, pressing Num Pad 0 will display Camera View (what the Camera sees). Pressing Num Pad 0 with Scene.001 selected has no effect since there is no Camera in Scene.001. There is also no Light in Scene.001, therefore, there will be no lighting effects until a Light is added to Scene.001.

If you wish to add items included in Scene to Scene.001 you simply copy and paste.

Fore example: To add the Camera and Light in Scene to Scene.001, in the 3D Viewport Editor, with **Scene** selected, hold Shift and LMB click on the Objects to be copied (Camera and Lamp) then RMB click and select Copy Objects in the menu that displays. Select Scene.001 in the Screen Header and in 3D Viewport Editor for Scene.001, RMB click and and select Paste Objects.

Note: In the Outliner Editor with **Scene** selected the display shows the Objects listed under **Collection** which is listed under **Scene Collection**. With **Scene.001** selected the display shows the Objects listed under **Scene Collection** (no sub entry Collection).

Figure 2.26

For details of how Objects are listed in Collections see Chapter 12.1 **Collections in the Outliner**.

2.9 Headers Menus and Panels

The default Blender Screen is made up with four Editor Panels and three Headers as seen in Figure 2.27. Each Editor Panel has a Header across the top of the Panel.

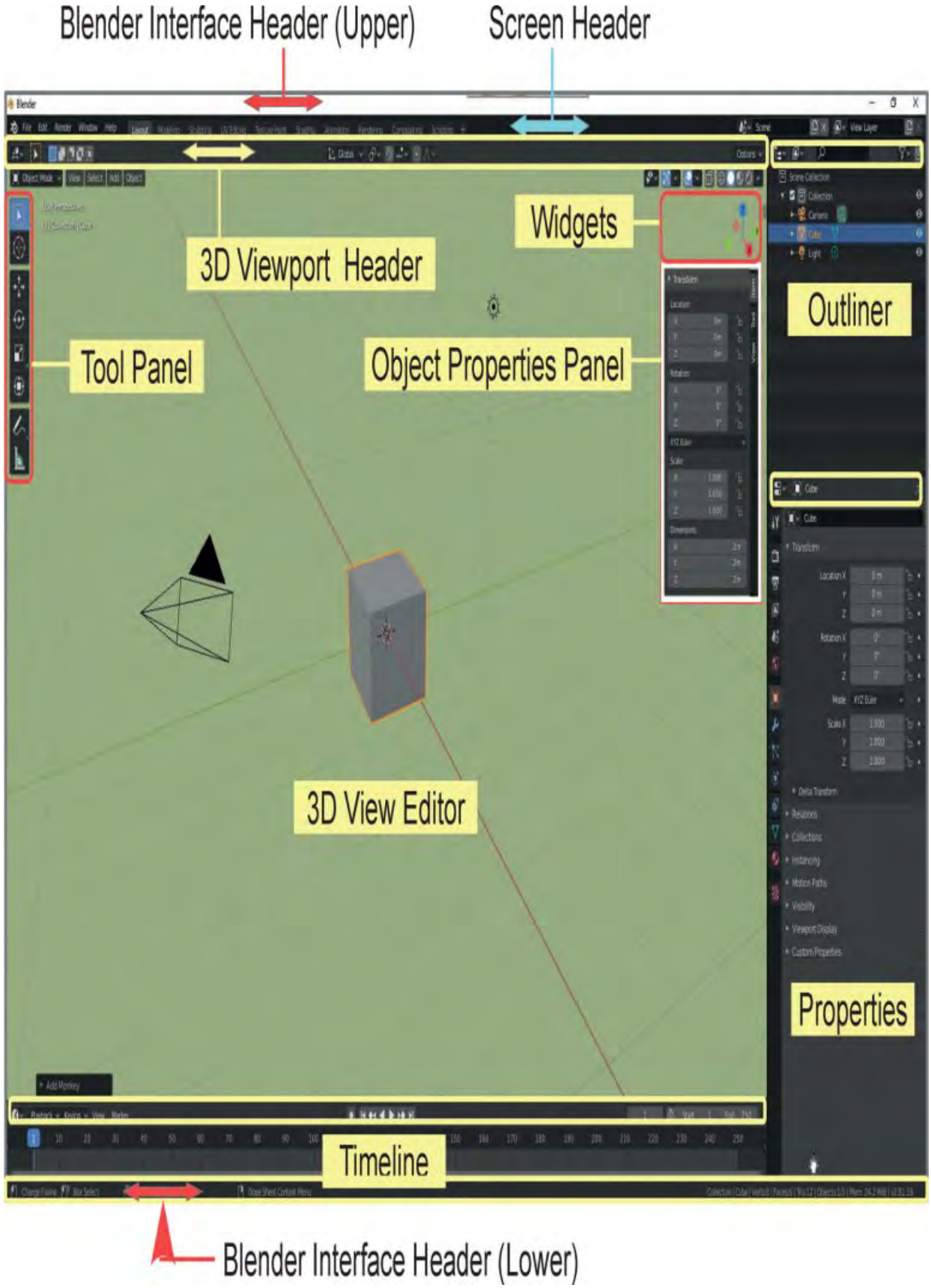

Blender Interface Header (Upper)
Screen Header
3D Viewport Header
Widgets
Outliner
Tool Panel
Object Properties Panel
3D View Editor
Properties
Timeline
Blender Interface Header (Lower)

Figure 2.27

The following diagrams will make you aware of the selection menus available in the Headers.

2.10 Headers and Panels

The Headers contain icons and text notations which are buttons that display Selection Menus containing options for activating functions affecting what takes place in the Editor. The options are numerous, therefore, it is not recommended that you attempt to memorise every option. Being aware of the different menus will help when reference is made in instructions. Specific references to selections will be made as required.

2.11 The Blender Interface Header

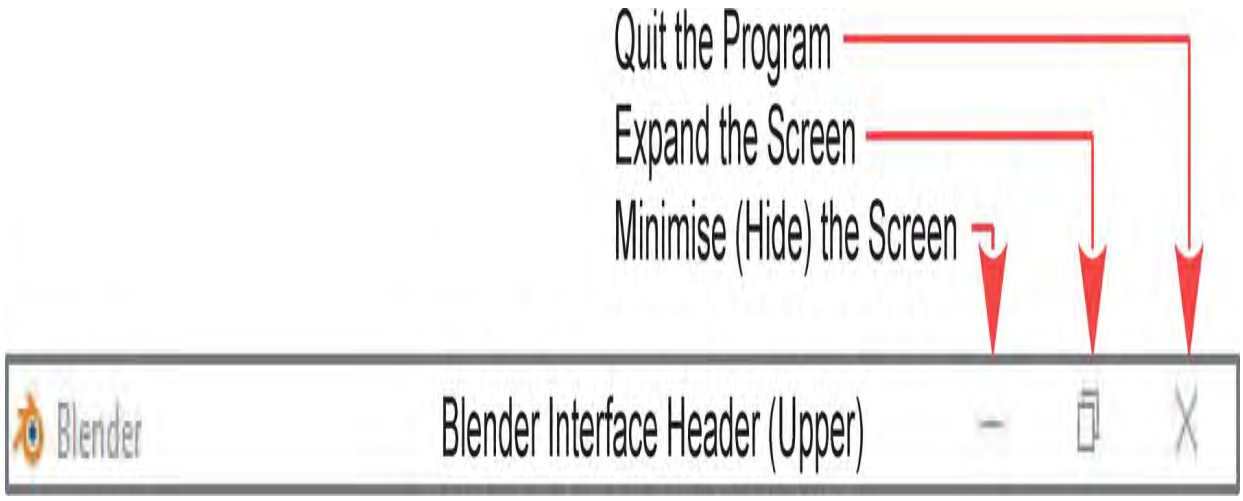


Figure 2.28

2.12 The Blender Screen Header

Click a Button to Display a Menu
Pre-Arranged **Workspaces** (Screen Arrangements)
File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Scripting
Splash Screen
Support Blender
About
Install Application Template...
New Ctrl N
Open... Ctrl O
Open Recent Shift Ctrl O
Revert
Recover
Save Ctrl S
Save As... Shift Ctrl S
Save Copy...
Link...
Append...
Data Previews
Import
Export
External Data
Defaults
Quit Ctrl Q
Undo Ctrl Z
Redo Shift Ctrl Z
Undo History...
Repeat Last Shift R
Repeat History...
Adjust Last Operation... F9
Operator Search... F3
Rename Active Item... F2
Lock Object Modes
Preferences...
New Scene
New
Copy Settings
Linked Copy
Full Copy
New Window
New Main Window
Toggle Window Fullscreen
Next Workspace Ctrl Page Down
Previous Workspace Ctrl Page Up
Show Status Bar
Save Screenshot
Toggle System Console
Render Image F12
Render Animation Ctrl F12
Render Audio...
View Render F11
View Animation Ctrl F11
Display Mode
Lock Interface
Manual
Tutorials
Support
User Communities
Developer Community
Python API Reference
Report a Bug
Save System Info
Blender 3.0 Manual
blender
Search docs

Figure 2.29

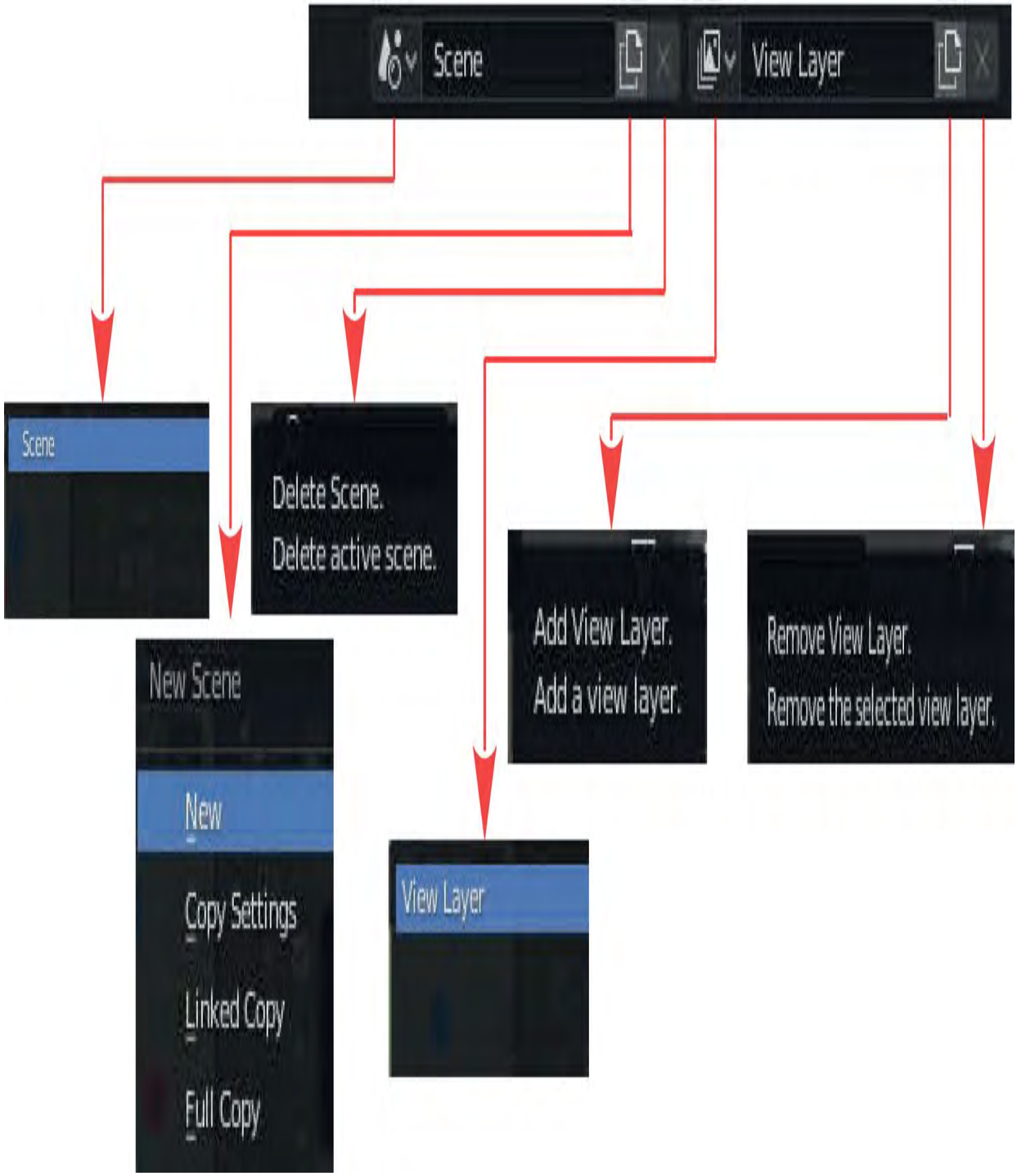


Figure 2.30

2.13 The 3D Viewport Header

The 3D Viewport Header is divided into three sections.

Note: The Header controls in Figure 2.31 are for the 3D Viewport when in **Object Mode**. The selection menus shown are a sample only.

The controls change when selecting an alternative Mode for the 3D Viewport Editor.

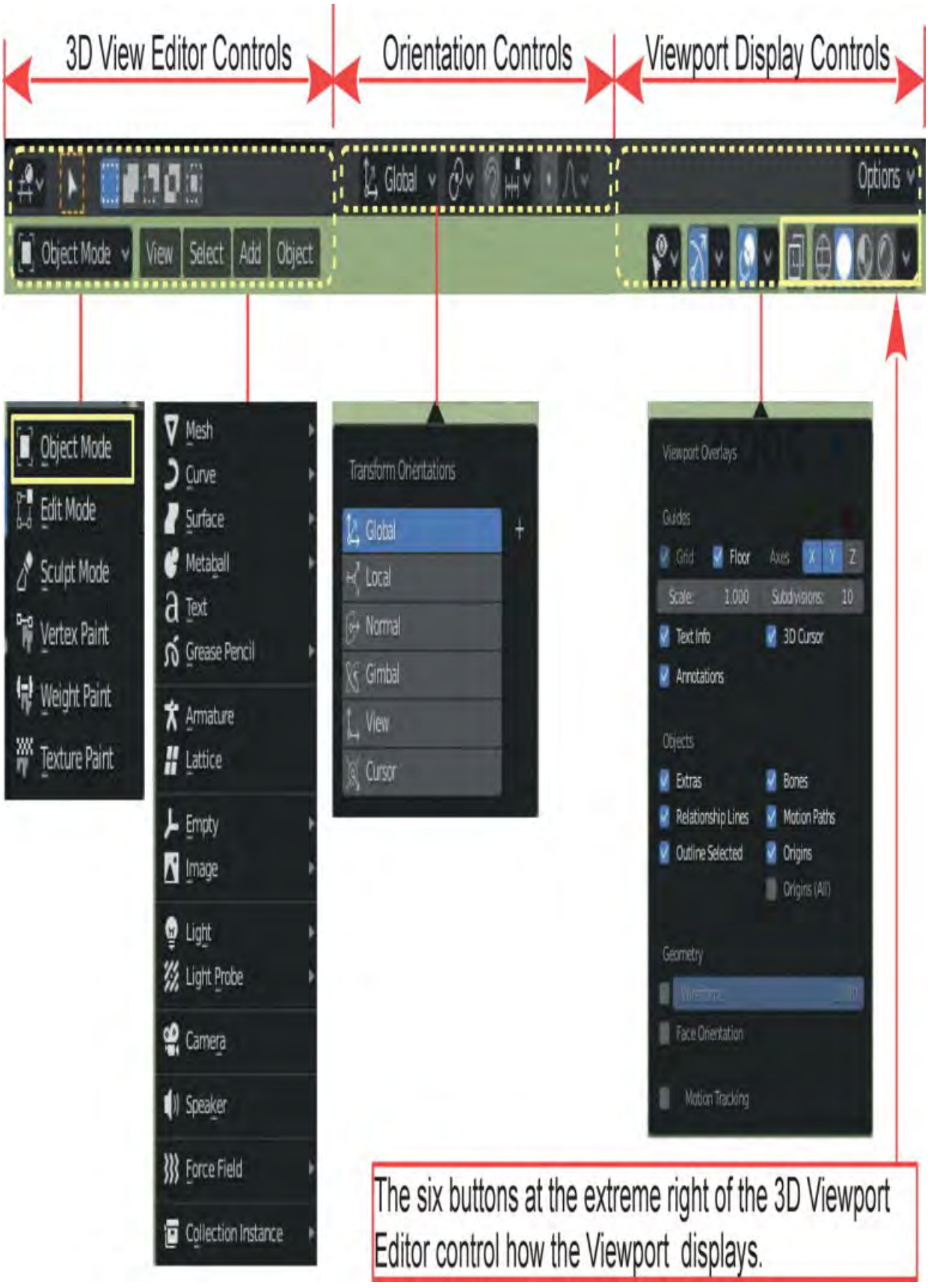

3D View Editor Controls
Orientation Controls
Viewport Display Controls
Global
Options
Object Mode
View
Select
Add
Object
Object Mode
Edit Mode
Sculpt Mode
Vertex Paint
Weight Paint
Texture Paint
Mesh
Curve
Surface
Metaball
Text
Grease Pencil
Armature
Lattice
Empty
Image
Light
Light Probe
Camera
Speaker
Force Field
Collection Instance
Transform Orientations
Global
Local
Normal
Gimbal
View
Cursor
Viewport Overlays
Guides
Grid
Floor
Axes
X
Y
Z
Scale:
1.000
Subdivisions:
10
Text Info
3D Cursor
Annotations
Objects
Extras
Bones
Relationship Lines
Motion Paths
Outline Selected
Origins
Origins (All)
Geometry
Face Orientation
Motion Tracking
The six buttons at the extreme right of the 3D Viewport Editor control how the Viewport displays.

Figure 2.31

2.14 The Widget Panel (See 2.7)

The Widget Panel is not a panel as such, but rather a collection of View Manipulation Tools (**Widgets**).

For a full description see section 2.

Mouse over on a Widget for a **description of its function**.

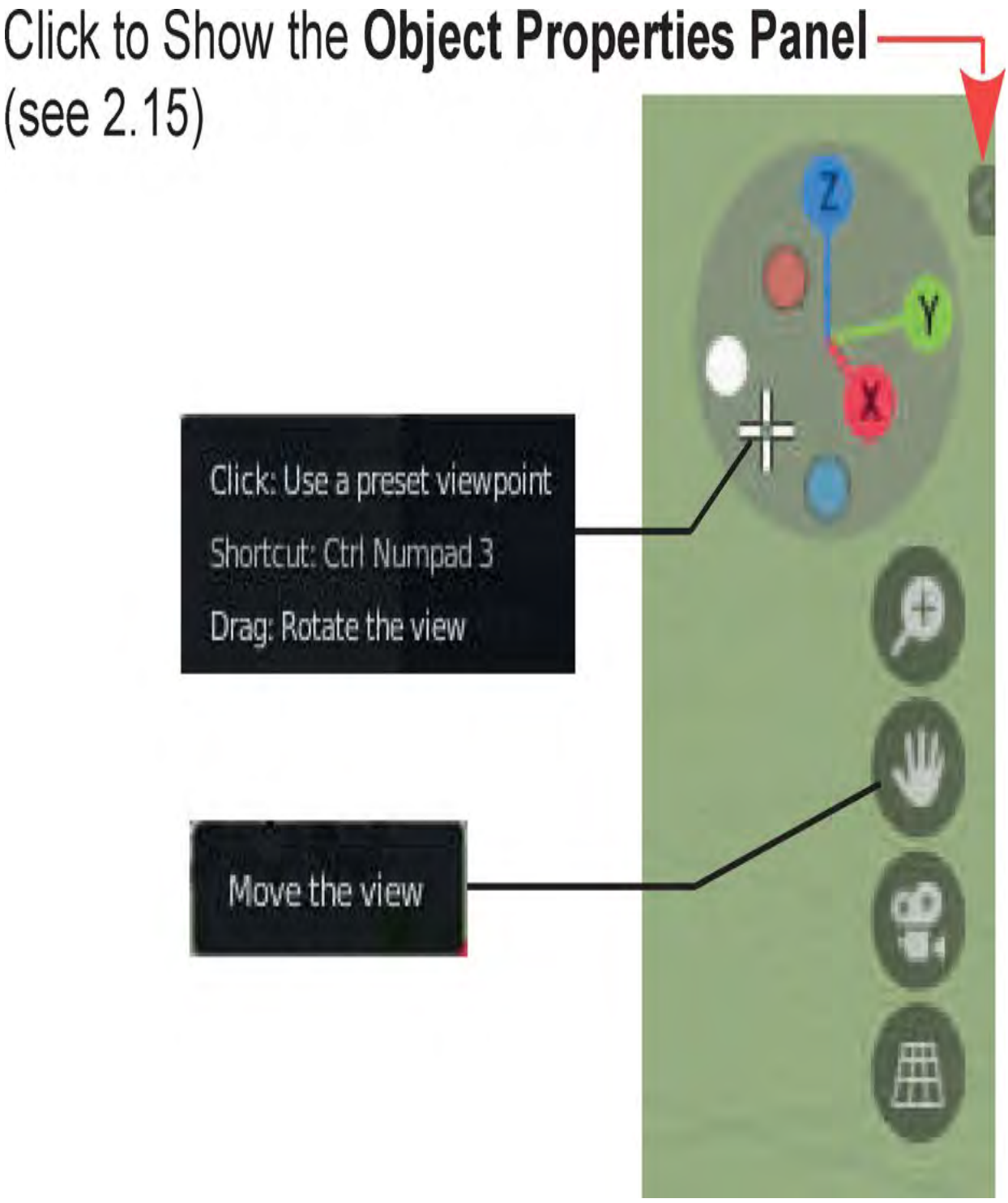


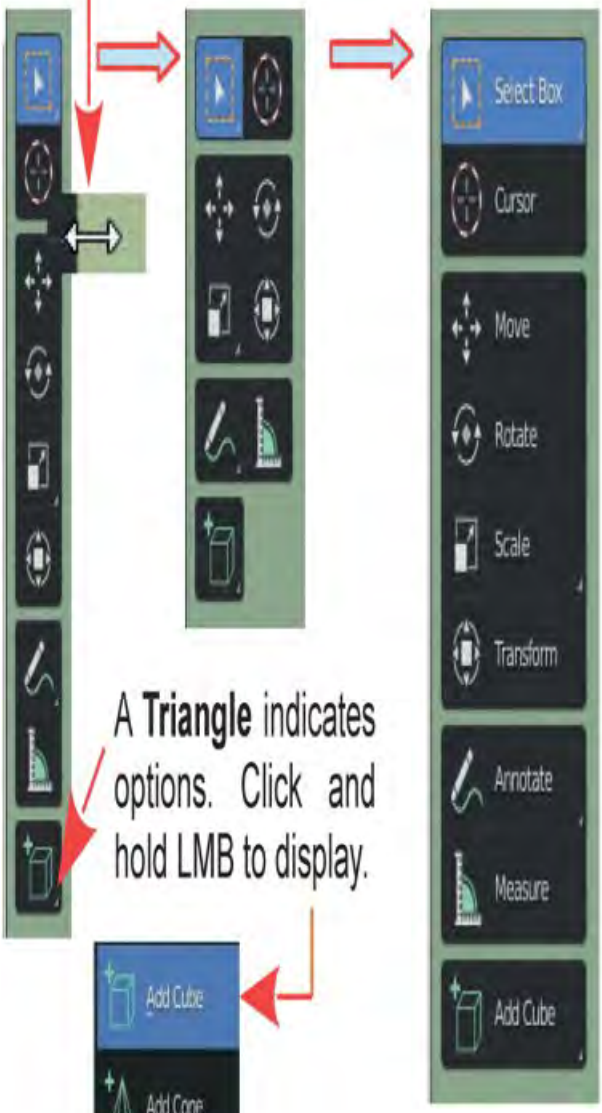
Figure 2.32

2.15 The Tool Panel and Panel

The Tool Panel at the left hand side of the 3D Viewport Editor houses Tools for performing operations on the Object which is selected. The Panel has three display arrangements.

Note: The content of the panel varies depending on which Mode is selected for the 3D Viewport Editor

Mouse over on the edge of the panel, click, hold and drag to increment the displays.



A **Triangle** indicates options. Click and hold LMB to display.

Object Properties have three display Modes. Click a Tab to change Modes

Press the **N Key** to show the panel.

Figure 2.33

The Object Properties Panel is hidden from view by default to conserve space in the 3D Viewport Editor.

Note: The **Object Properties Panel** allows you to adjust values affecting the Properties of the selected Object. For example increasing the X Location value in the positive direction moves the Object on the X Axis in the 3D Viewport Editor.

Note: To hide the Object Properties Panel press the **N Key.** You may show and hide the **Tool Panel** by pressing the **T Key**. Alternatively click on or Chevrons in upper corners of the Editor when panels are hidden.

2.16 Properties Editor Tabs

The Properties Editor is the panel below the Outliner Editor which extends to the bottom of the Screen. This Editor is the engine room for Blender containing controls for actions in the 3D Viewport Editor. The controls will be explained as you progress through the book and encounter the different features of the program. What you see in the Properties Editor depends on which selection button is activated (Figure 2.34 over).

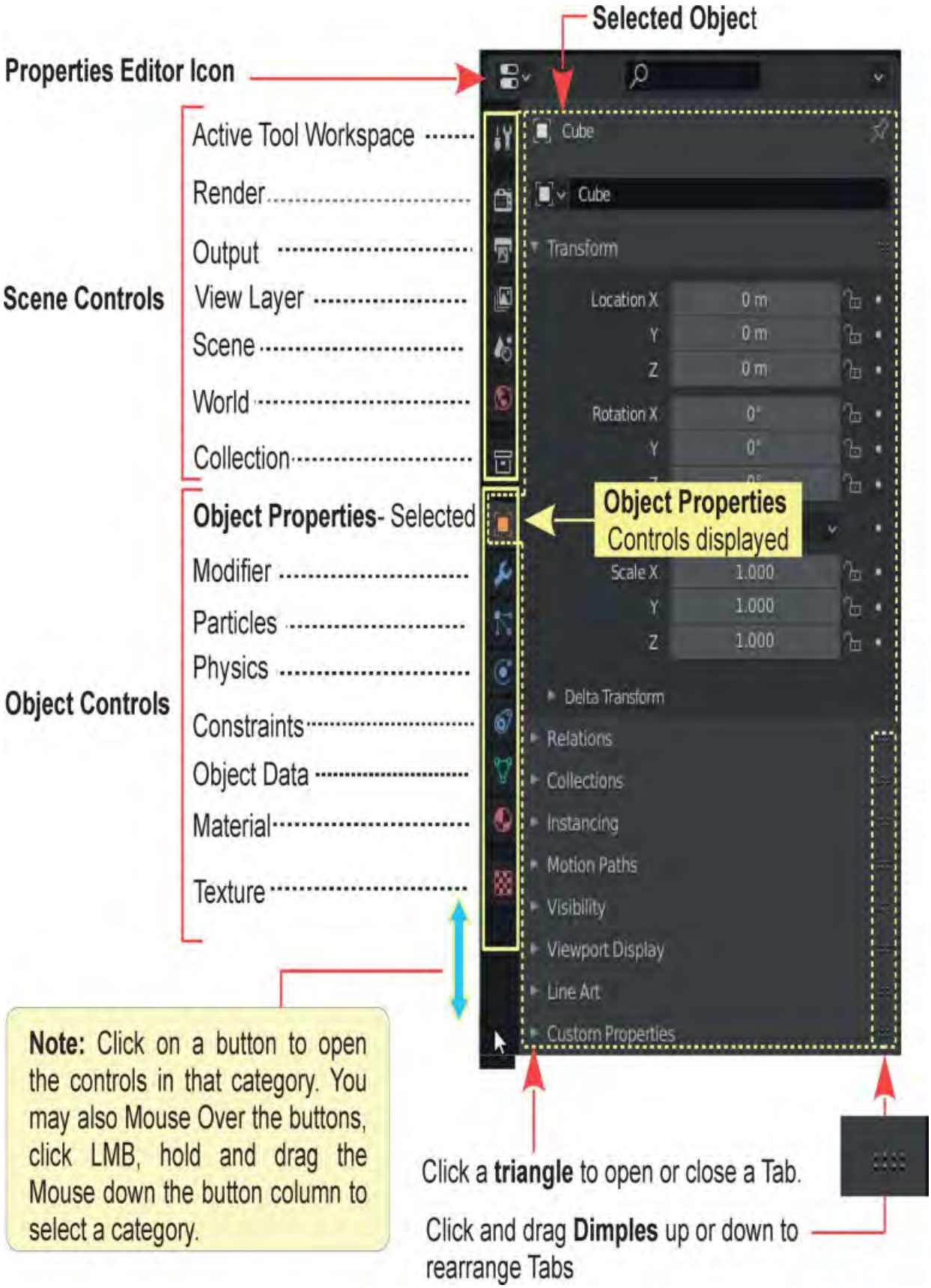

Selected Object
Properties Editor Icon
Active Tool Workspace
Render
Output
Scene Controls
View Layer
Scene
World
Collection
Object Properties- Selected
Modifier
Particles
Physics
Object Controls
Constraints
Object Data
Material
Texture
Cube
Cube
Transform
Location X
Y
Z
Rotation X
Y
0 m
0 m
0 m
0°
0°
Object Properties
Controls displayed
Scale X
Y
Z
1.000
1.000
1.000
Delta Transform
Relations
Collections
Instancing
Motion Paths
Visibility
Viewport Display
Line Art
Custom Properties
Note: Click on a button to open the controls in that category. You may also Mouse Over the buttons, click LMB, hold and drag the Mouse down the button column to select a category.
Click a triangle to open or close a Tab.
Click and drag Dimples up or down to rearrange Tabs

Figure 2.34

There are fifteen buttons from which to choose, each will present a different display of controls. The buttons are arranged in two categories (Scene Controls and Object Controls).

Figures 2.34 shows the Properties Edit with the **Object Properties** selected. The Object buttons control the properties of **the Object that is selected** in the 3D Viewport Editor Scene. The default Object is the Cube. The Object button displays controls and information about the Object which you have **selected** in the 3D Viewport Editor.

2.17 The Preferences Editor

The **Preferences Editor** is a behind scenes control panel for configuring and adding functionality to Blender. The controls are extensive, therefore, will be referenced as the various features are required while progressing through the book. At this point you should be aware of the Editors existence and have a general understanding of what it entails.

The **Preferences Editor** may be opened by clicking an **Editor Icon** and selecting **Preferences** in the Editor selection menu (Figure 2.1). This method will fill the entire Editor Panel. For example selecting the 3D Viewport Editor icon and selecting Preferences fills the 3D Viewport Editor Panel with the Preferences Editor.

For many operations opening the Preferences Editor in this way is fine but when configuring the 3D Viewport Editor from the Preferences Editor it is advantageous to see changes as they are made.

Access the Preferences Editor from the Blender Screen Header.

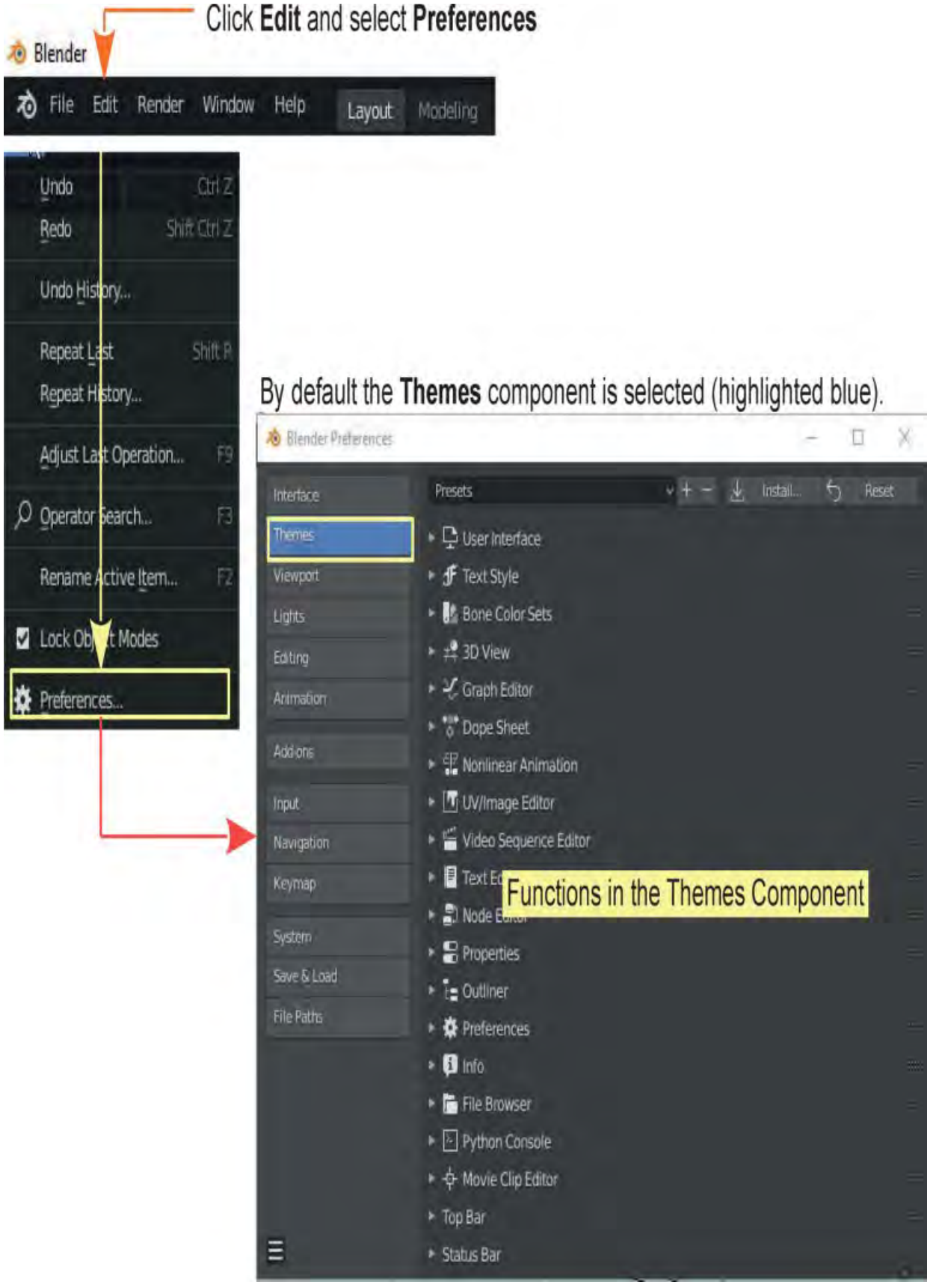

Click **Edit** and select **Preferences**
Blender
File Edit Render Window Help Layout Modeling
Undo Ctrl Z
Redo Shift Ctrl Z
Undo History...
Repeat Last Shift R
Repeat History...
Adjust Last Operation... F9
Operator Search... F3
Rename Active Item... F2
Lock Object Modes
Preferences...
By default the **Themes** component is selected (highlighted blue).
Blender Preferences
Interface
Themes
Viewport
Lights
Editing
Animation
Add-ons
Input
Navigation
Keymap
System
Save & Load
File Paths
Presets
Install...
Reset
User Interface
Text Style
Bone Color Sets
3D View
Graph Editor
Dope Sheet
Nonlinear Animation
UV/Image Editor
Video Sequence Editor
Properties
Outliner
Preferences
Info
File Browser
Python Console
Movie Clip Editor
Top Bar
Status Bar
Functions in the Themes Component

Figure 2.35

Opening the **Preferences Editor** using this method opens the Editor in a panel overlaid in the upper left hand corner of the Screen. The Editor has a component selection menu at the left hand side and a list of the functions included in the selected component at the right.

By default the**Themes**component is selected (highlighted blue).

Note: Although the Themes option is the default display, if you have used a different option and closed the Preferences Editor, whatever option you selected will display next time you open.

2.18 3D Viewport Editor - Background Color

In the **Preferences Editor, Themes**, click the triangle adjacent to **3D View**. With the **Mouse Cursor** in the right hand panel of the **Preferences Editor**, scroll down, **way way down**, until you see **Theme Space**. Expand this entry (click the triangle) and scroll way way down again to **Gradient Colors**. Expand this entry. You will find the **Gradient High/Off** color bar. This is where you change the color of the 3D Viewport Editor background.

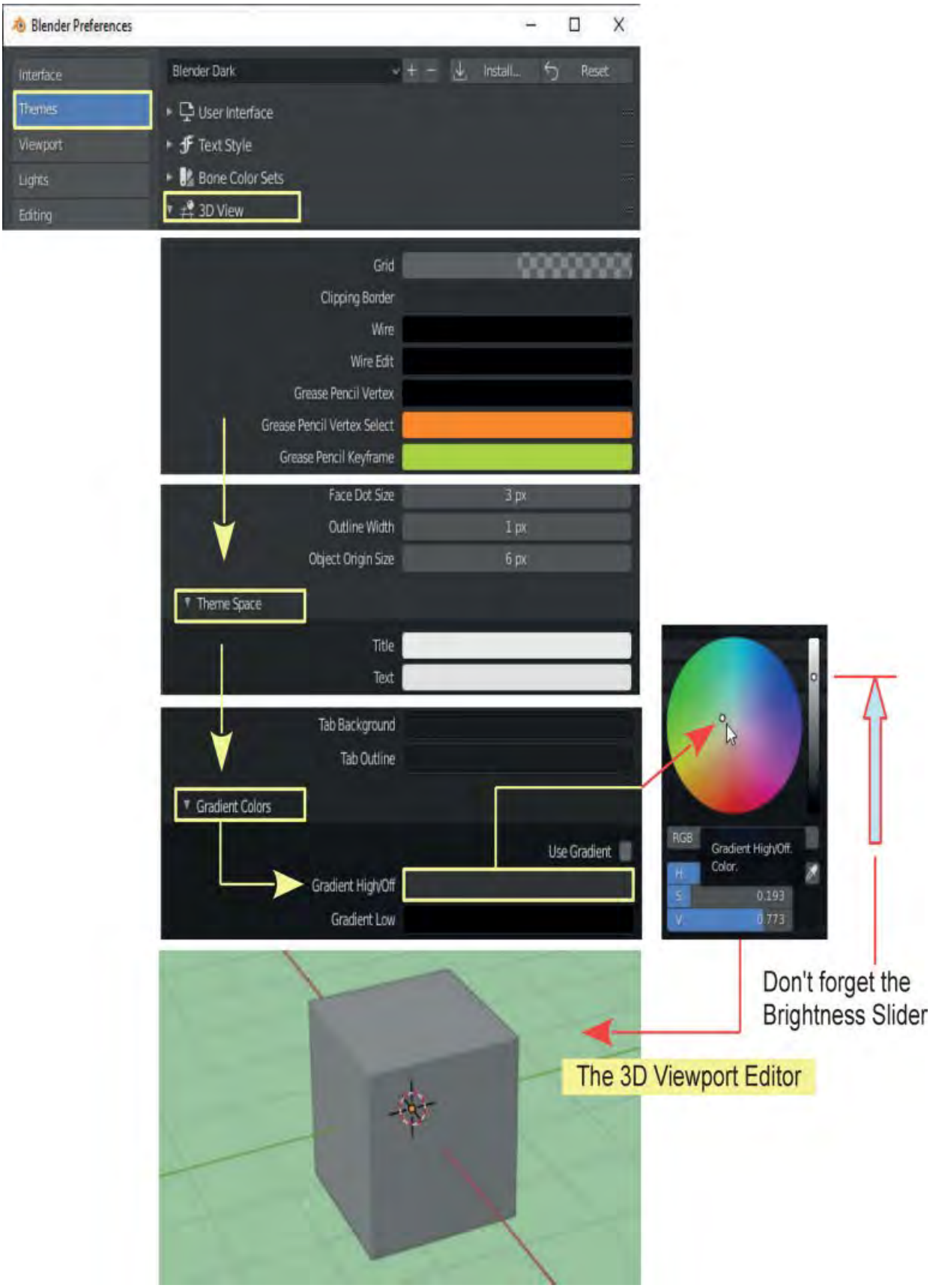

Blender Preferences
Interface
Themes
Viewport
Lights
Editing
Blender Dark
Install...
Reset
User Interface
Text Style
Bone Color Sets
3D View
Grid
Clipping Border
Wire
Wire Edit
Grease Pencil Vertex
Grease Pencil Vertex Select
Grease Pencil Keyframe
Face Dot Size
3 px
Outline Width
1 px
Object Origin Size
6 px
Theme Space
Title
Text
Tab Background
Tab Outline
Gradient Colors
Use Gradient
Gradient High/Off
Gradient Low
RGB
Gradient High/Off. Color.
H
S
0.193
V
0.773
Don't forget the Brightness Slider
The 3D Viewport Editor

Figure 2.36

2.19 Workspaces

Workspaces are the arrangement of Editor panels configured for specific working procedures.

The default arrangement of Editors is configured to provide a space for compiling a Scene, that is primarily to enter and position Objects. The Objects may be for creating models or they could be pre-constructed models of characters. The default arrangement is named the **Layout Workspace**.

In the default Screen arrangement Header you will see **Layout** highlighted white which indicates that this Workspace is the **Active Workspace**.

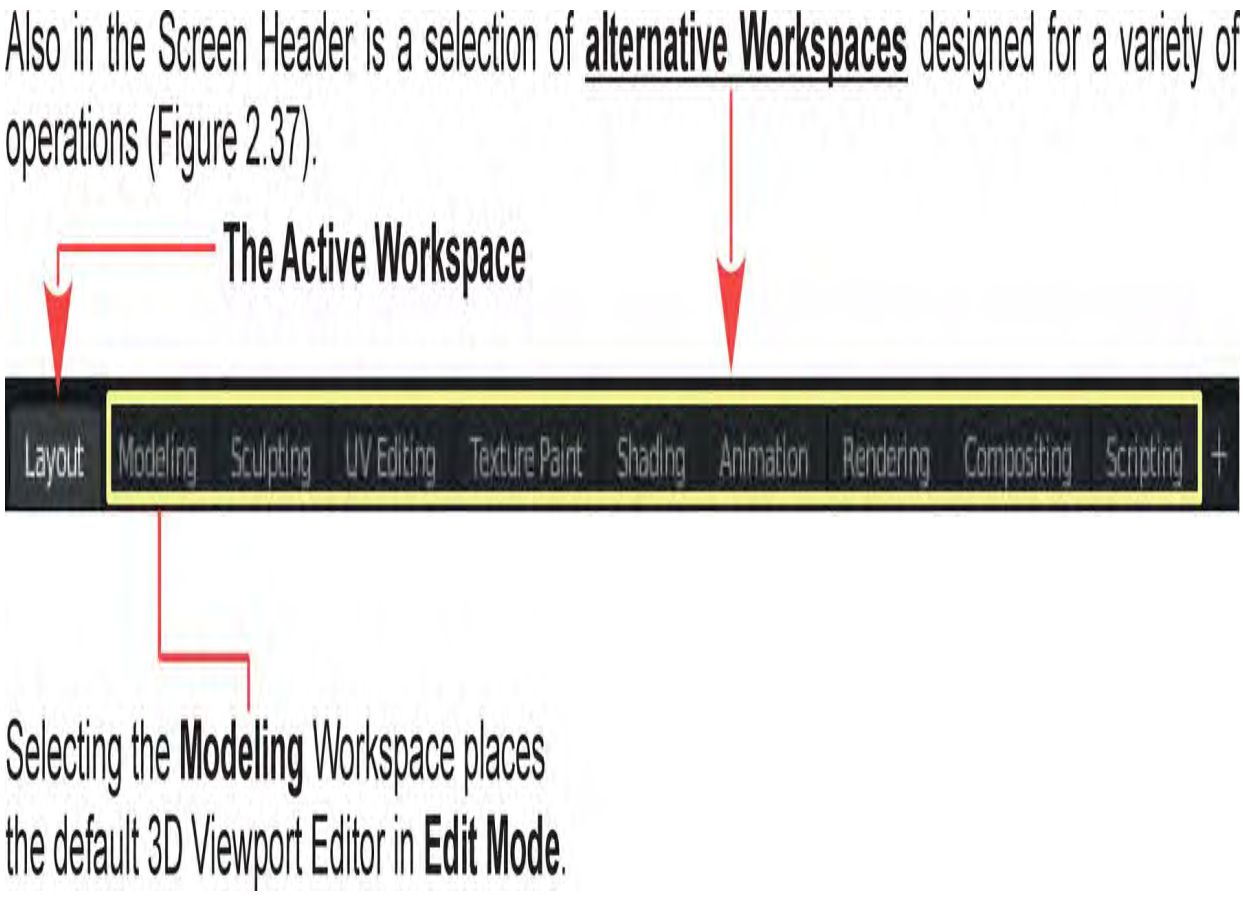


Figure 2.37

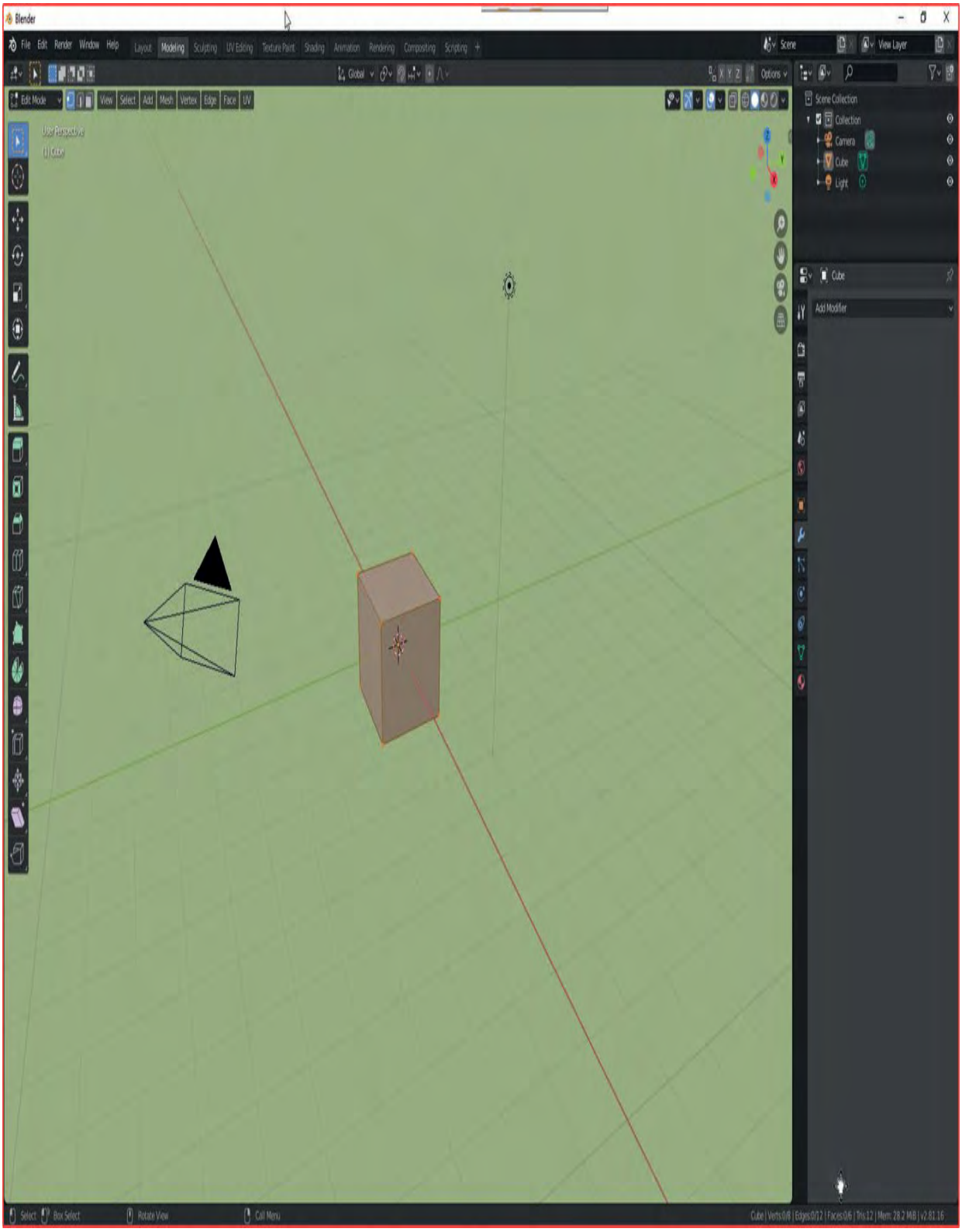

Blender
File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Scripting
Scene
View Layer
Edit Mode
View Select Add Mesh Vertex Edge Face UV
Global
Options
Scene Collection
Collection
Camera
Cube
Light
Add Modifier
Select
Box Select
Rotate View
Call Menu

Figure 2.38

2.20 Creating New Workspaces

As explained in Section 2.19, **Workspaces** are the arrangement of Editor Panels configured for specific working procedures. You may select one of the **Pre-Assembled Workspace** in the Blender Screen Header (Figure 2.39 over) or you can create a new unique Workspace.

Figure 2.39

Click the cross in the Blender Screen Header at the RHS of the Workspace options. Click **Duplicate Current** in the menu that displays. Duplicating Current generates a copy of the current Screen arrangement and automatically names it **Layout.001**. The new name is displayed in the Header.

Click on **Layout.001** in the Blender Screen Header to ensure you have Layout.001 displayed in the Screen. You may now reconfigure the Screen arrangement to suit your working requirements. Rearrange the Blender Screen

by splitting Editors and moving borders and changing the Editor Types. This creates a new Workspace (Figure 2.40).

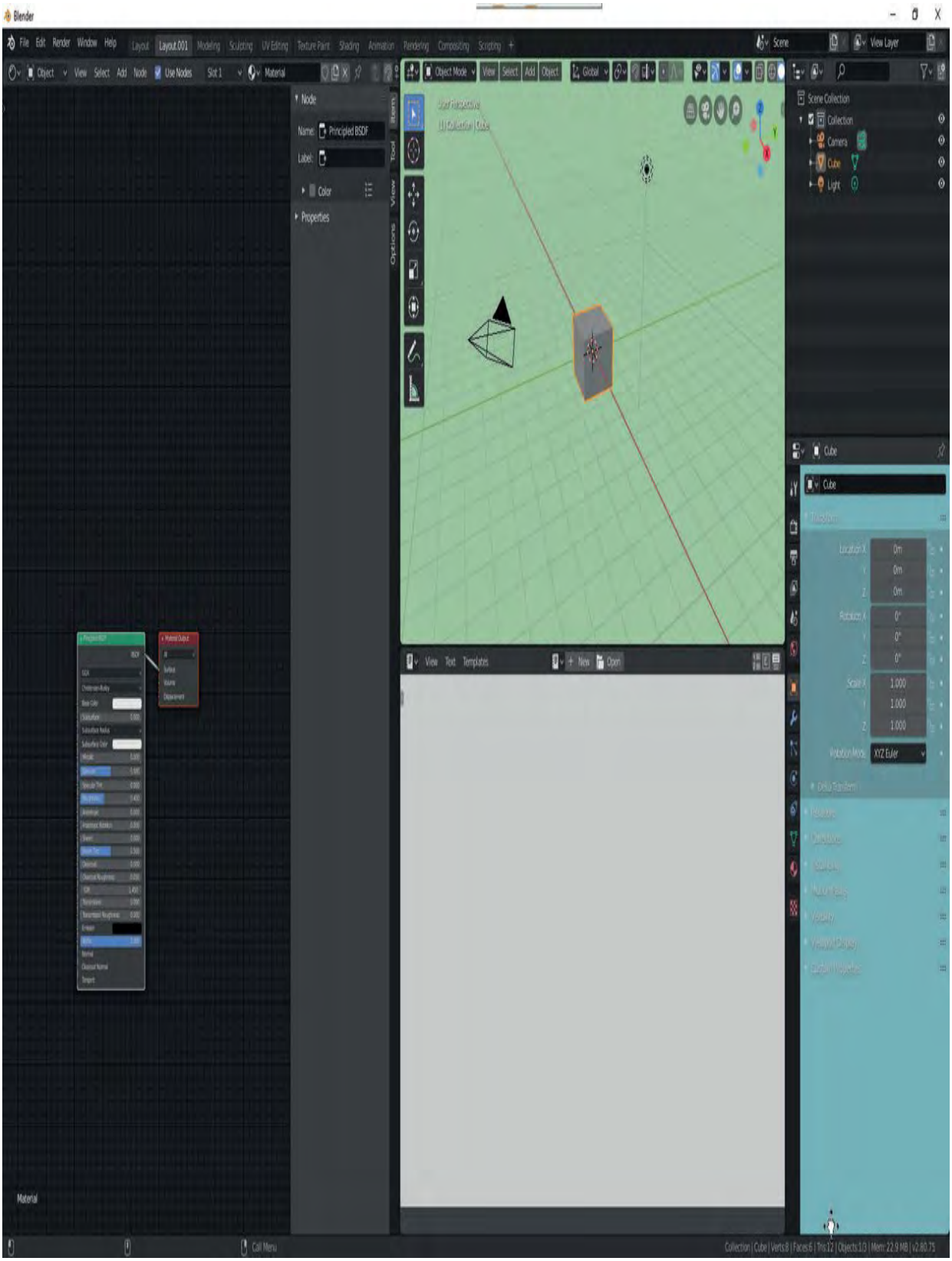

Blender
File Edit Render Window Help
Layout Layout.001 Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Scripting
Scene
View Layer
Object View Select Add Node Use Nodes Slot 1 Material
Node
Name: Principled BSDF
Label:
Color
Properties
Object Mode View Select Add Object Global
Scene Collection
Collection
Camera
Cube
Light
Cube
Location X 0m
Rotation X 0°
Scale X 1.000
Rotation Mode XYZ Euler
View Text Templates New Open
Material
Call Menu
Collection | Cube | Verts:8 | Faces:6 | Tris:12 | Objects:1/3 | Mem: 22.9 MB | v2.80.75

Figure 2.40

Note: The new Workspace is only available in the current Blender File. If the current file is the default Blender file, save the current file with a new name to make the new Workspace available for future work.

With **Layout.001** configured, selecting **Layout** in the Screen header changes the Screen back to the default Screen arrangement.

Renaming Workspaces

You may create as many Workspaces as you like by repeating the procedure outlined above. New Workspaces will be named: Layout.001, Layout.002, Layout.003, Layout.004 etc.

You rename a Workspace by double clicking on the name in the Screen Header, backspace and retype a new name.

Saving a Workspace

Note: Creating a Workspace (Working Environment) in the default Blender Screen arrangement is, in fact, modifying the default Blender File. This procedure occurs whenever you change anything in the default Screen. This means, whenever you model something, change a color or create an animation, the changes are, therefore, only available in the Blender file that is open when the changes are made.

To use changes in future work you have to save the Blender File, as a new file, then reopen it when you want to use something contained in the file. This applies to Workspaces.

How to save a Blender file is discussed in Chapter 3.

2.21 Themes

A **Theme** is the cosmetic appearance of the interface. This is purely a matter of taste and personal preference. Working in a pleasing environment has a

beneficial effect on work output and changing the environment occasionally can give a new lease of life.

For your convenience Blender has a choice of nine Themes. You may download and install new Themes into Blender from the Internet.

Inbuilt Themes: are found in the **Preferences Editor**. You can change an Editor Panel to the Preferences Editor as demonstrated in Section 2.1 but for convenience, when changing the appearance of the Blender interface, it is better to open the Preferences Editor from the Blender Screen Header. This method opens the Editor as a small separate Panel over the top of the GUI allowing you to see changes as the occur .

Figure 2.41

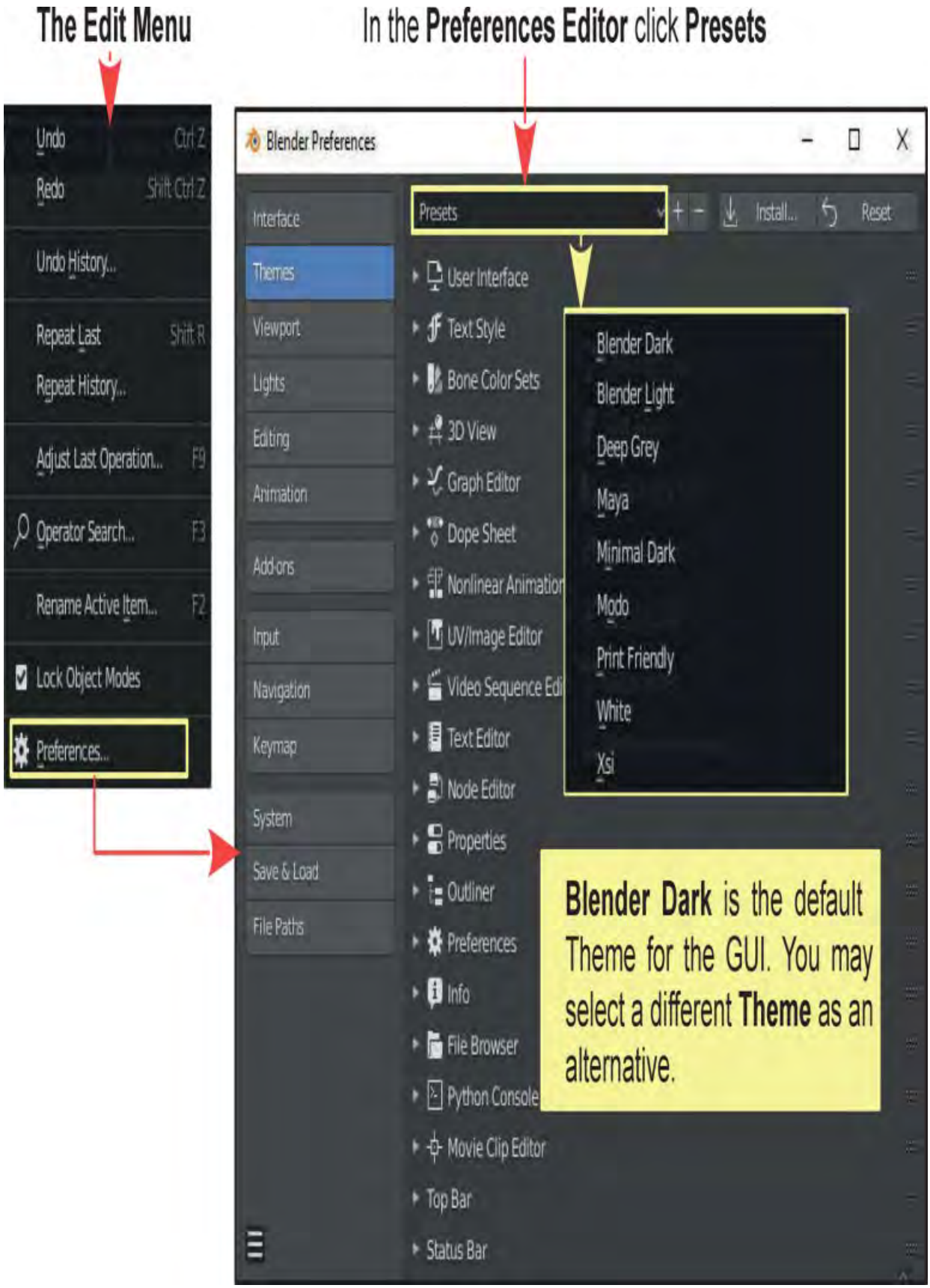

The Edit Menu
In the Preferences Editor click Presets
Undo Ctrl Z
Redo Shift Ctrl Z
Undo History...
Repeat Last Shift R
Repeat History...
Adjust Last Operation... F9
Operator Search... F3
Rename Active Item... F2
Lock Object Modes
Preferences...
Blender Preferences
Interface
Themes
Viewport
Lights
Editing
Animation
Add-ons
Input
Navigation
Keymap
System
Save & Load
File Paths
Presets
Install...
Reset
User Interface
Text Style
Bone Color Sets
3D View
Graph Editor
Dope Sheet
Nonlinear Animation
UV/Image Editor
Video Sequence Edi
Text Editor
Node Editor
Properties
Outliner
Preferences
Info
File Browser
Python Console
Movie Clip Editor
Top Bar
Status Bar
Blender Dark
Blender Light
Deep Grey
Maya
Minimal Dark
Modo
Print Friendly
White
Xsi
Blender Dark is the default Theme for the GUI. You may select a different Theme as an alternative.

Figure 2.42

Close the Preferences Editor.

When Blender is closed and reopened the selected Theme will be displayed.

Installing New Themes

You may download and install themes from the Internet. Themes are downloaded as Blender

2.81 XML Files. One example is found at:

Https://blenderartists.org/t/theme-awsome-theme-for-blender-2-8/1120656

Figure 2.43

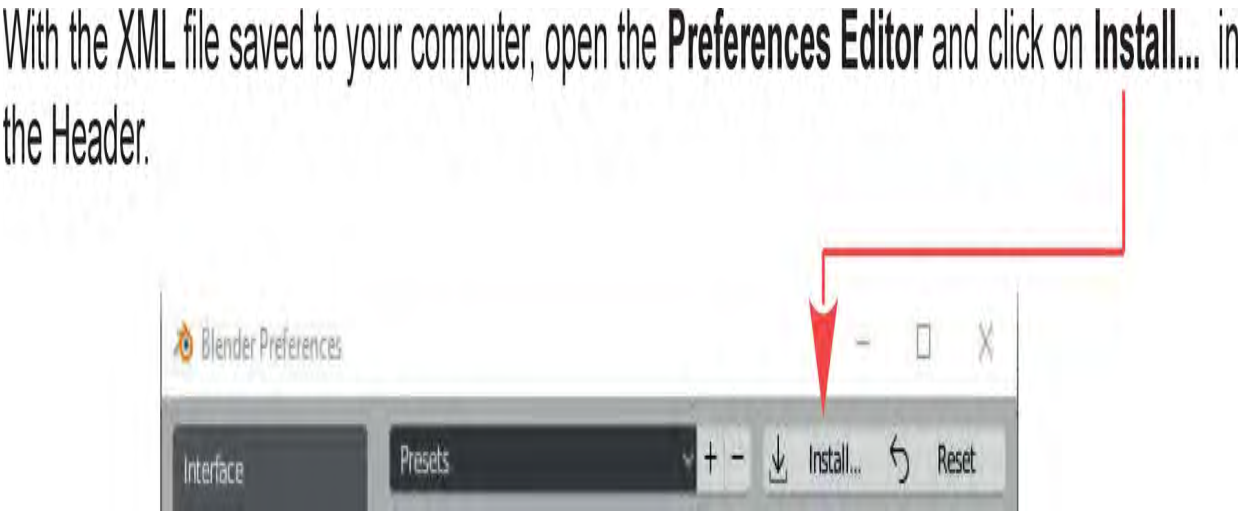


Figure 2.44

Navigate to the **XML file** in the **File Browser** (see Chapter 3), select the file and click **Install Theme**.

The **Awesome** Theme will be available in the **Theme Preset** menu.

Figure 2.45

Note: Saving Files and Navigating the **Blender File Browser** is explained in Chapter 3.

Creating New Themes

A new Theme may be created from within Blender. In Figure 1.1 in Chapter 1 the background colors of the Editor Panels were modified to make it easier to identify the individual Panels. These modification can be saved as a Theme for future use.

To change Editor background colors, open the **Preferences Editor** and have **Themes** selected in the left hand column. Click on a triangle adjacent to the Editor you wish to modify, for example the **Properties Editor**, then click the triangle adjacent to **Theme Space**. You will see Window Background and a gray color bar. Click the color bar and select a new color in the color picker.

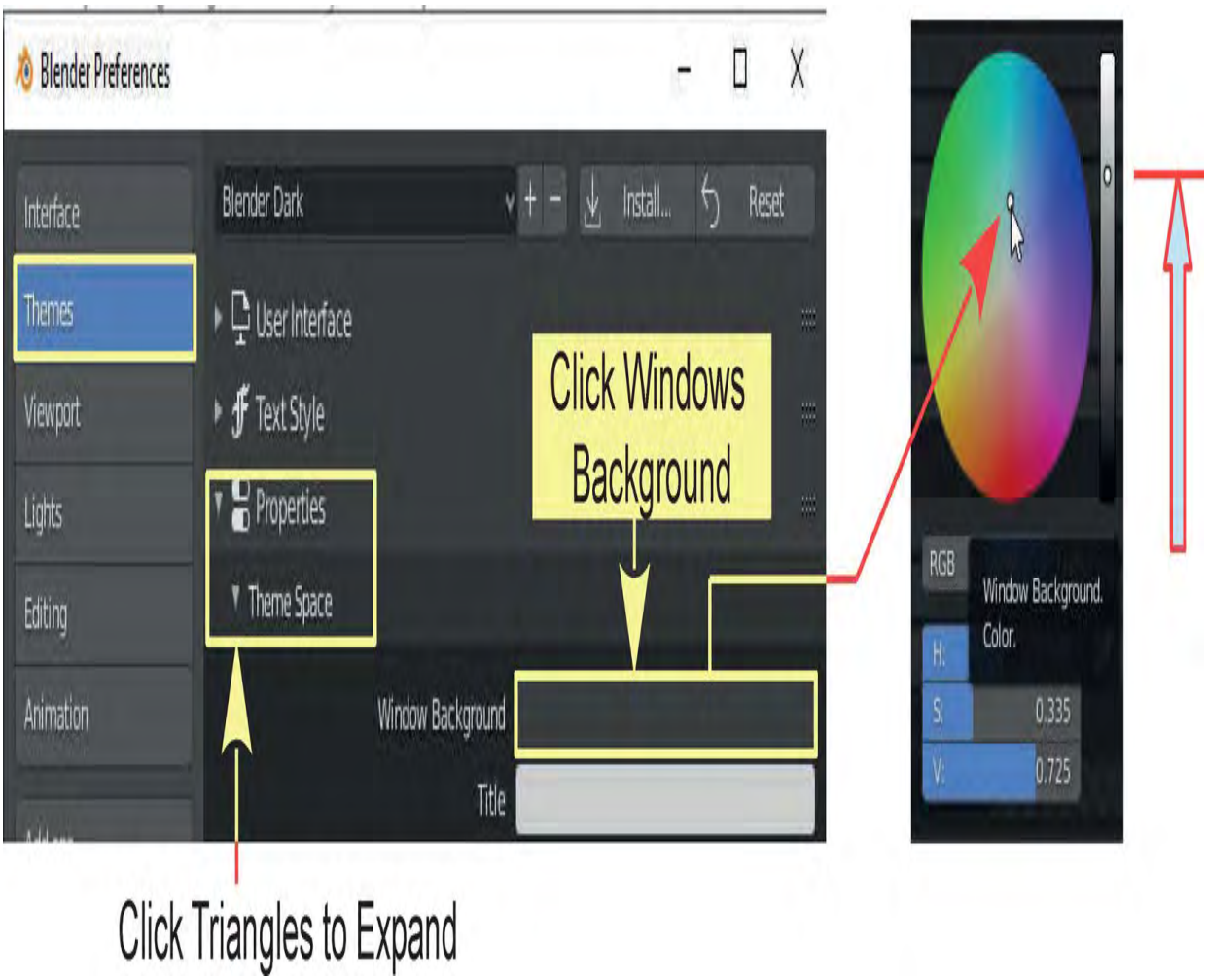


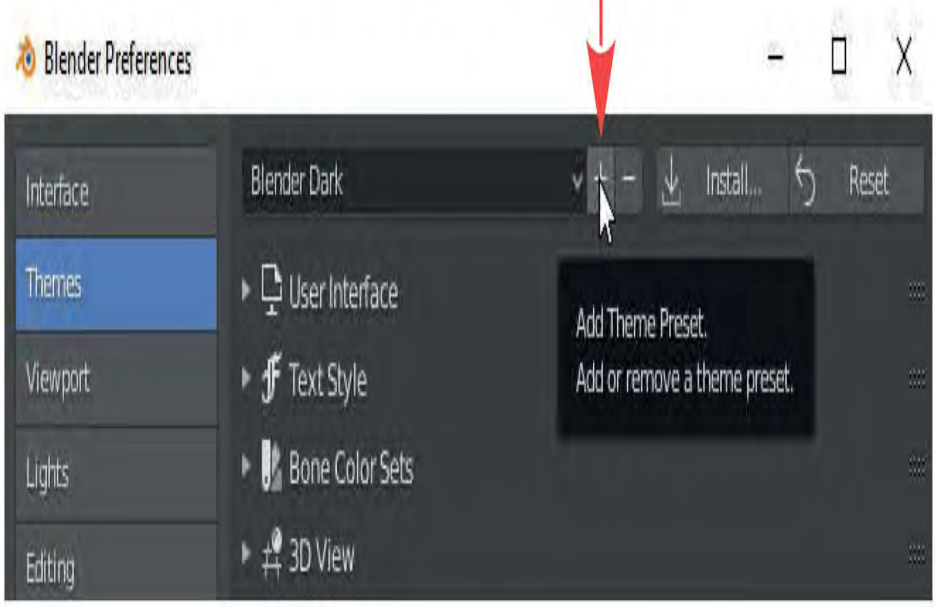
Figure 2.46

Selecting a color changes the color in the Color Bar and the background of the Editor Panel.

You may repeat this procedure for any Editor Type but be aware, the 3D Viewport Editor different, as explained in section 2.18. Changing the color of the 3D Viewport Editor is, in creating a new Theme.

2.22 Saving a Theme

In the **Preferences Editor** Header click on **Add Theme Preset**.



My New Theme is entered as one of the available Themes.

Figure 2.47

In the panel that displays click in the area next to **Name** to show a typing cursor and type a name for the Theme. Click **OK**.

Note: A Theme, including a new Theme, may be applied to any Workspace. As with Themes, you can create your own unique Workspaces.

3 Navigate and Save

c3.1 Files and Folders
c3.2 Saving a File
c3.3 Window File Explorer
c3.4 Windows File Explorer Diagram
c3.5 Blender File Browser
c3.6 Opening Files
c3.7 File Browser Editor Features
c3.8 Make a New Folder
c3.9 Display Options
c3.10 Saving Your Work
c3.11 The Concept of Files
c3.12 The Append or Link Command
c3.13 Importing Objects
c3.14 Activating Import File Types
c3.15 Packing Data

Saving Work

When you work in Blender you edit (modify) the default file which opens when you start Blender or a file that has been previously saved. Blender file names end with a **.blend** suffix and are peculiar to the Blender program. Saving work means you save the modifications or editing, that has been performed in a Blender file. You save the file, in a folder of your choice on your computers hard drive. You should understand how and where to create a folder and how to retrieve a file when it has been saved. In other words you

need to know how to navigate your file system. In Blender files are saved on your computer using the **File Browser Editor.**

Navigation

Navigation is the science of finding the way from one place to another. If you can see where you are going its an easy process to head over to that place but sometimes where you want to go is hidden from view. In Blender you create files and store them away for future use. You can reuse the files and build on to them and then save the new material. Saved files are your library of information from which you extract elements and insert into future work. The saying is, **"There is no point reinventing the wheel".** If you have created something that works use it again. But where did you put the wheel? That's where navigation comes in. You need to find the place where you safely stored that wheel or, in the case of Blender, where you saved a file containing the wheel. Navigation in Blender is performed in the **File Browser Editor.**

3.1 Files and Folders Definition (from the Internet)

A file is a common storage unit in a computer, and all programs and data are "written" into a file and "read" from a file. A folder holds one or more files, and a folder can be empty until it is filled. A folder can contain other folders (sub folders). Folders provide a method for organising files much like a Manila file folder contains paper documents in a file cabinet. In fact, files that contain text are often called documents.

Folders are also called "directories," and they are created on the hard drive (HD) or solid state drive (SSD) when the operating system and applications are installed. Files are always stored in folders. In fact, even the computer desktop is a folder; a special kind of folder that displays its contents across the entire screen.

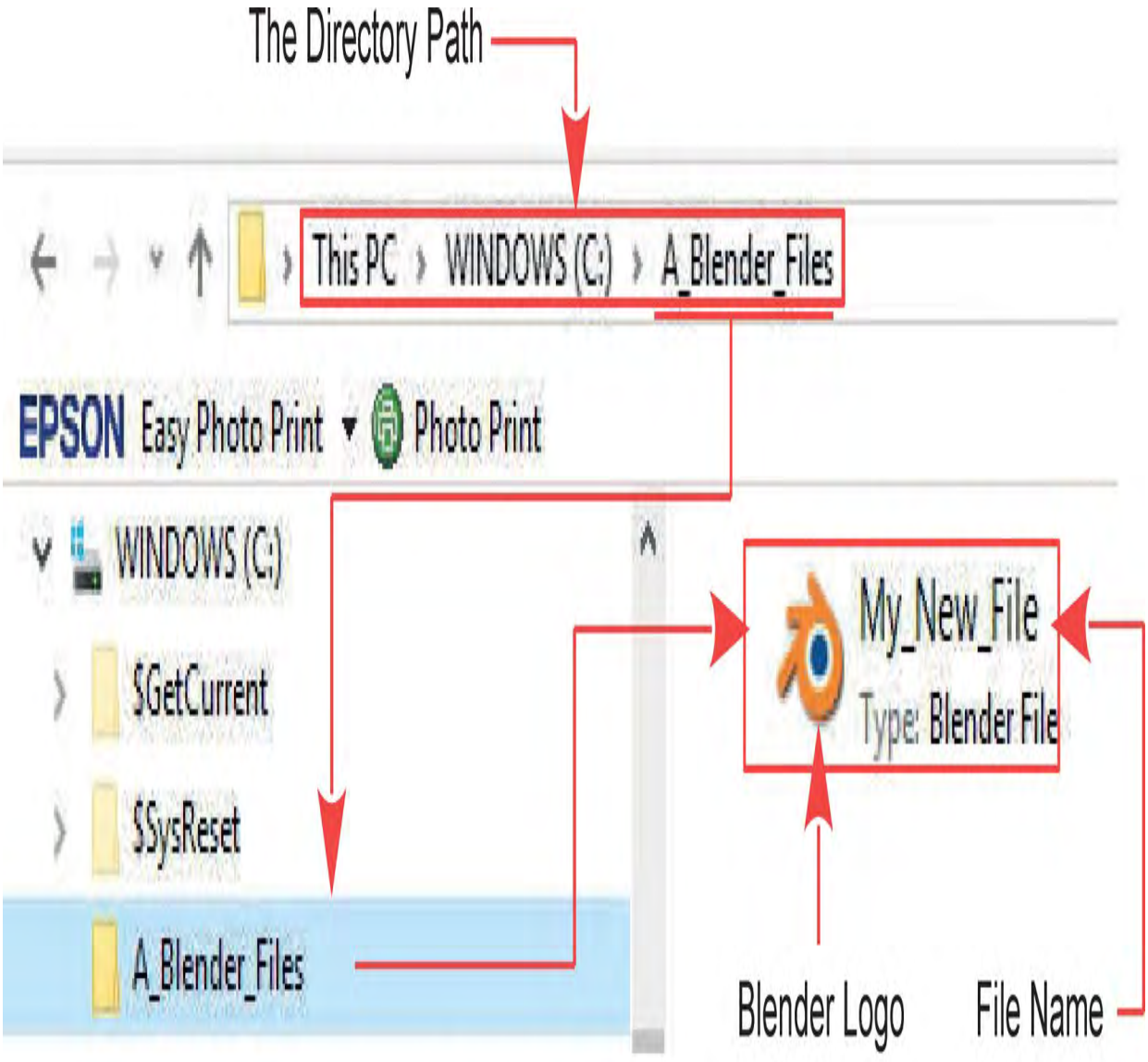
File Extensions

A file extension or suffix, is the bit at the end of a file name preceded by a dot or period. For example; My_Photo.JPEG, would be a JPEG image (photograph). The .JPEG extension tells the computer which application (App) or program to use when opening the file. With a .JPEG extension the computer would look for an image editor or viewer to open the file. With a .TXT extension, signifying a text file, the computer would use a text editor.

When writing file extensions to a file name they are usually written in lower case letters such as **.jpeg** or .**txt**.

Blender files have a **.blend** extension which tells the computer to open the file in the Blender program.

Figure 3.1 show a Blender file saved in the **C: Directory** (Hard Drive) in a Folder named **A_Blender_Files.** The Blender file is named **My_New_File.** Blender file names usually display with the Blender logo preceding the file name but the **.blend** file extension does not always display.



Windows 10 File Explorer

Figure 3.1 show a Blender file saved in the **C: Directory** (Hard Drive) in a Folder named **A_Blender_Files**. The Blender file is named **My_New_File**. Blender file names usually display with the Blender logo preceding the file name but the **.blend** file extension does not always display.

3.2 Saving a File

On a computer, when you save a Blender file (.blend) you are saving the data which is producing the display on the computer screen. This set of data includes not only what you see but all the settings which control all the effects

that will be displayed in the various Editors. The Blender file may be considered as a complete package. Saving a file for the default arrangement saves everything.

Figure 3.2 shows the data listed in the Outliner Editor which would be saved for the default Blender Scene. Even before any editing has taken place the list is extensive. All this data is saved to a single file. ‘Figure 3.2

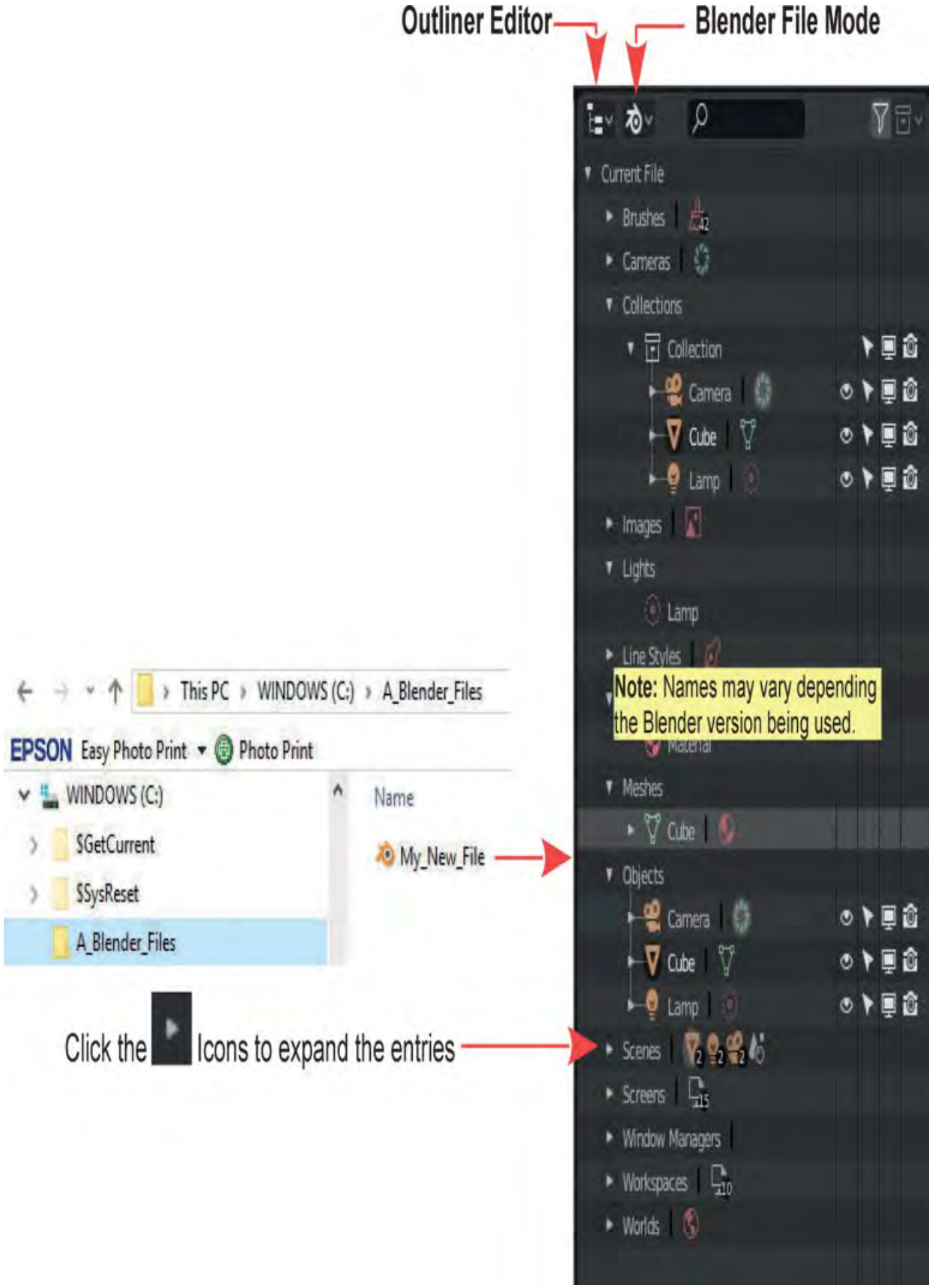

Outliner Editor
Blender File Mode
Current File
Brushes
Cameras
Collections
Collection
Camera
Cube
Lamp
Images
Lights
Lamp
Line Styles
Note: Names may vary depending the Blender version being used.
Meshes
Cube
Objects
Camera
Cube
Lamp
Scenes
Screens
Window Managers
Workspaces
Worlds
This PC › WINDOWS (C:) › A_Blender_Files
EPSON Easy Photo Print
Photo Print
WINDOWS (C:)
$GetCurrent
$SysReset
A_Blender_Files
Name
My_New_File
Click the
Icons to expand the entries

Figure 3.2

The diagram shows the file saved on the **C:** Drive (Hard Drive) of the computer in the Directory Folder named **A_Blender_Files.** The file has been named **My_New_File.** Although not shown, in this Windows file system the file does have a **.blend** suffix.

Note: Figure 3.2 shows the **Outliner Editor in Blender File Mode.**

Note: Placing an **A** in front of the Directory Folder name ensures that it is located at the top of the alphabetical directory list.

In Blender the **File Browser Editor** (Figure 3.3) is used to navigate through the file system on your computer. On a Windows operating system, **Windows Explorer** or **Windows File Explorer** are used. Blenders File Browser is a little different to the Windows system in appearance but uses the same work flow.

Blender File Browser
Blender [C:\A_Blender_Files\My_New_File.blend]
File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Scripting
View Select
Volumes
C:\A_Blender_Files\
WINDOWS (C:)
RECOVERY (D:)
DVD RW Drive (E:)
Name
My_New_File.blend

Figure 3.3

View
Select
Volumes
WINDOWS (C:)
RECOVERY (D:)
DVD RW Drive (E:)
System
Documents
Desktop
Favorites
Add Bookmark
Recents
Downloads
Name
$RECYCLE.BIN
$SysReset
A_Blender_6th_2.81_Otd
A_BTD_Components
A_BTD_Helens_Stuff
A_Nikon_Photos
A_PDF_Conversion
A_Photos
A_Russell_80
A_Russell_80_Pics
A_Russell_Reports
A_Submarine_Movie
Animation_Nodes_Blender
Blender_2.80
Blender_2.81_Beta
Blender_2.82_Alpha
Blender_4th_Edition_T
Blender_4th_Index_Compiler
Blender_4th_Suppliments_PDF

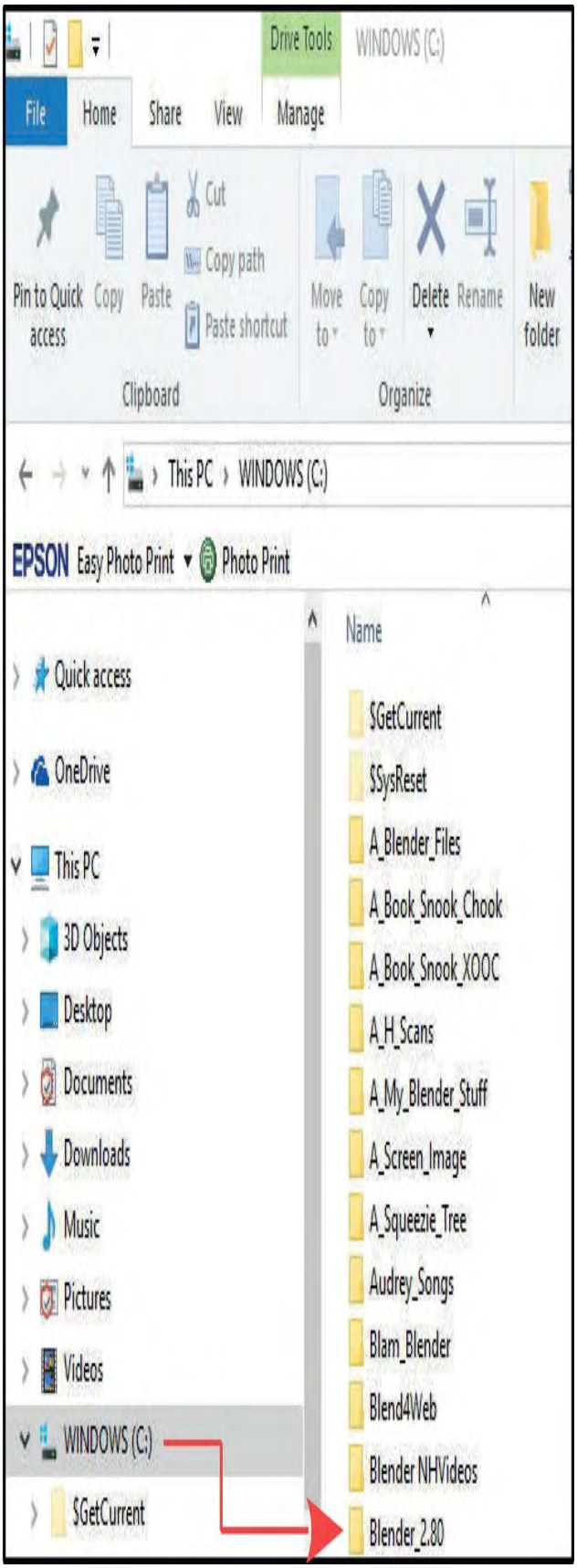

Drive Tools
WINDOWS (C:)
File
Home
Share
View
Manage
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
New folder
Organize
This PC › WINDOWS (C:)
EPSON Easy Photo Print
Photo Print
Quick access
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
WINDOWS (C:)
$GetCurrent
Name
$GetCurrent
$SysReset
A_Blender_Files
A_Book_Snook_Chook
A_Book_Snook_XOOC
A_H_Scans
A_My_Blender_Stuff
A_Screen_Image
A_Squeezie_Tree
Audrey_Songs
Blam_Blender
Blend4Web
Blender NHVideos
Blender_2.80

Figure 3.4

3.3 Windows File Explorer

Take a short refresher to analyse what you do when you save something when using a Windows operating system.

As an example, it is assumed you have written a note using **Windows Notepad** and you are about to save the file. You simply go to the top of the Notepad window, click on File, click on Save and the Save As window displays. In the panel at the top of this window you will probably see a panel showing This PC > Documents which is telling you that your file will be saved to the Documents folder on your computer (Figure 3.5). You enter a name for your file (My_Blender Note.bct) and click Save. Simple!

Windows Notepad – Save As

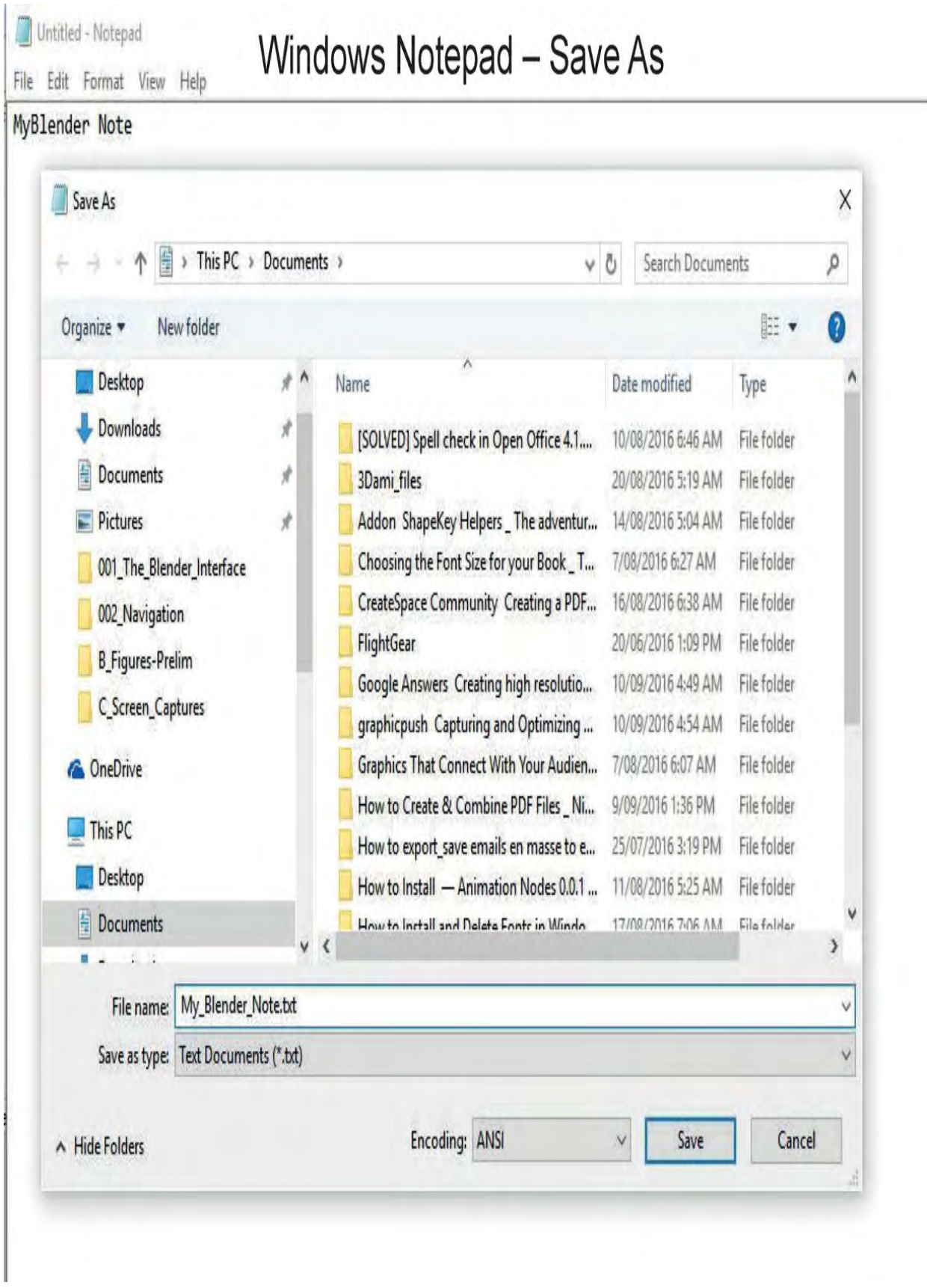
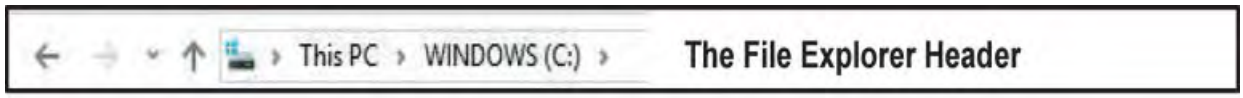


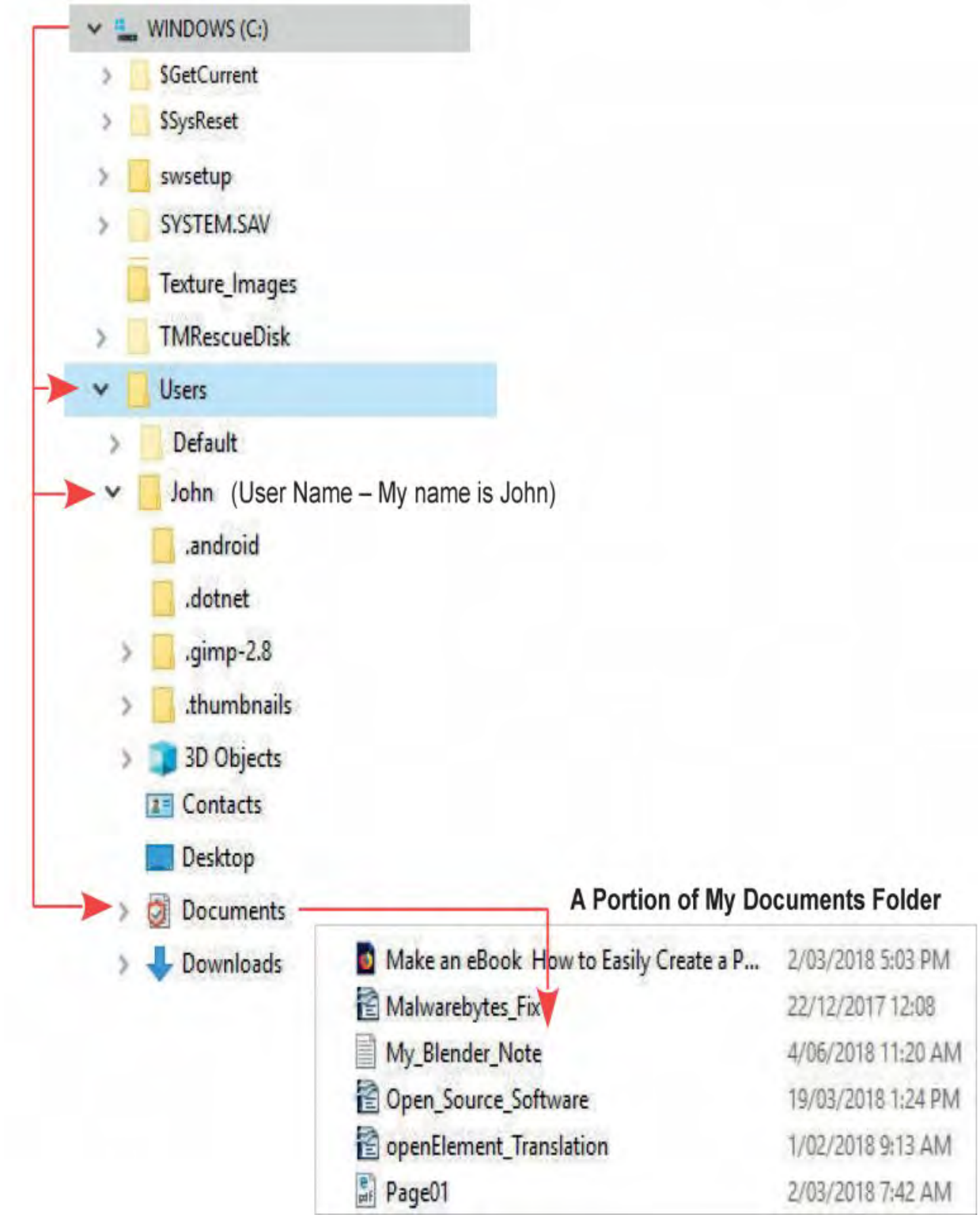
Figure 3.5

The problem with this is; the file gets saved amongst your letters to your mother, the tax man, pictures of your pet dog and holiday snaps all saved in the Documents folder. You should make a special folder for your **Blender Stuff.** You can create new folders in File Explorer by Right clicking on a folder or sub folder, selecting **New** then clicking **New Folder** and entering a name. You probably know how to do this but it is more important for you to understand how to do this is in Blender.

3.4 Windows File Explorer Diagram



The File Explorer Header



A Portion of My Documents Folder

Figure 3.6

3.5 Blender File Browse.

Blenders File Browser is the **File Browser Editor.** Click on the 3D Viewport Editor icon (upper left) and select **Editor Type: File Browser** in the menu. The 3D Viewport Editor is replaced by the **File Browser Editor** (Figure 3.8). This is where you navigate to find things and save things. You save your Blender files and rendered images and animation files and you search for pre-saved files from which you obtain data to use in new work. You can also create a new folder for your **Blender Stuff.**

To navigate the File Browser is very simple. As an example, go find the file named **My_Blender_Note.txt.** The .txt bit (suffix) on the end of the name tells you that the file is a Text file.

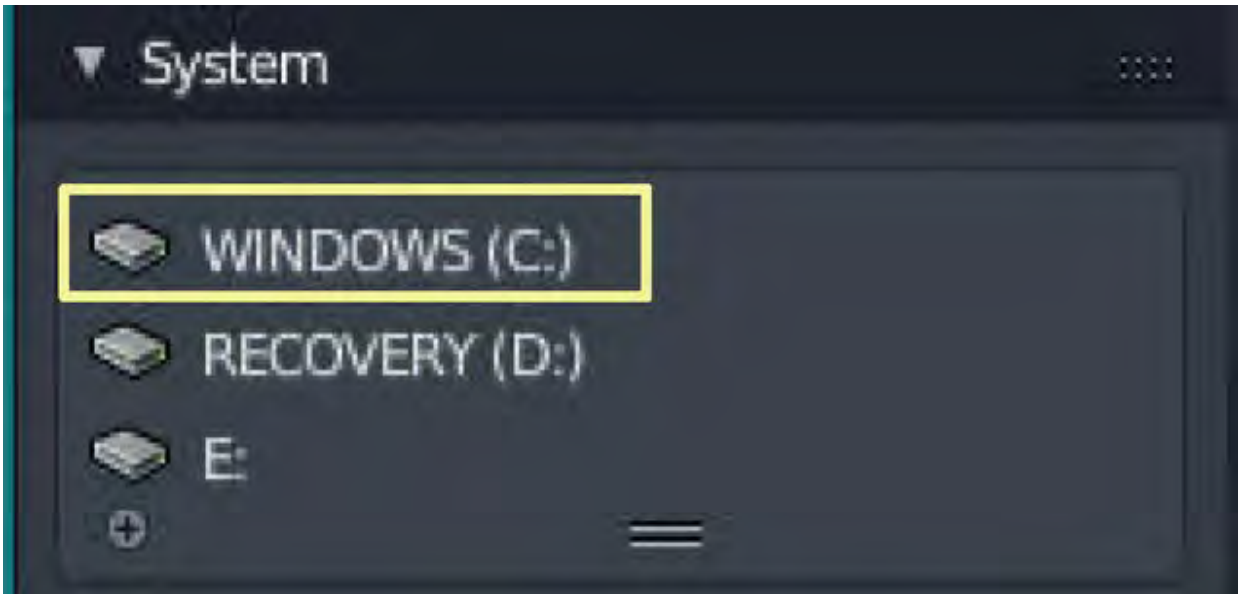


Figure 3.7

In the File Browser Editor click on **(C:)** or **Windows (C:)** in the **System Tab** panel in the upper LH part of the Editor panel. This is the C: drive on your PC. The PC used in this demonstration is a HP computer running

Windows 10. The name on your computer is probably different but you will have something that tells you it is your C: Drive.

When you click on the C: Drive the main RH panel (in the File Browser Editor) displays the list of folders that you have in the C: Drive. The list is displayed in columns and by default is in alphabetical order. To follow the file path that was shown in Windows File Explorer:

In each panel, click the heading (highlights blue or gray) then double click to open the next panel.

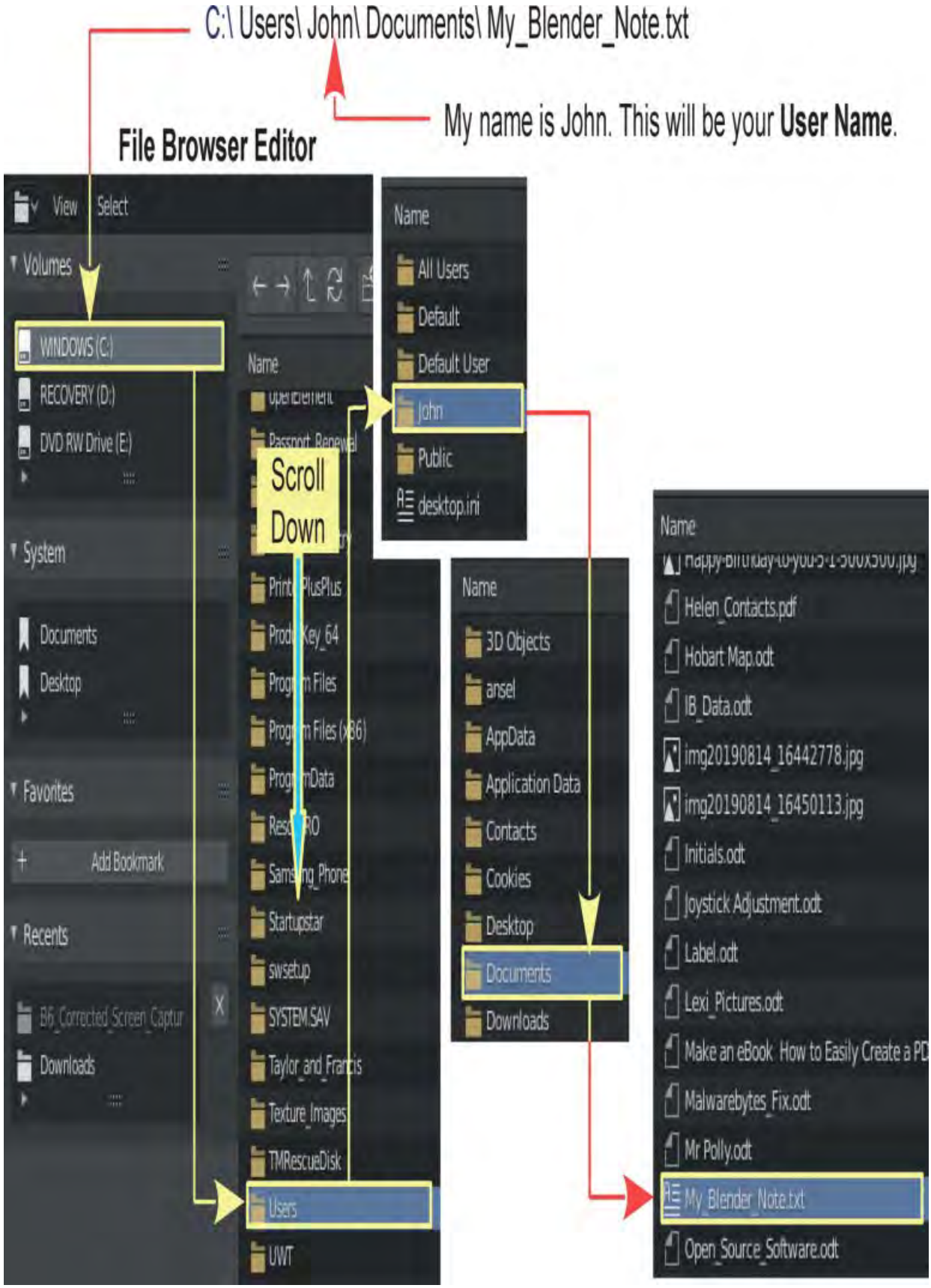

C:\ Users\ John\ Documents\ My_Blender_Note.txt
My name is John. This will be your **User Name**.
File Browser Editor
View
Select
Volumes
WINDOWS (C:)
RECOVERY (D:)
DVD RW Drive (E:)
System
Documents
Desktop
Favorites
Add Bookmark
Recents
Downloads
Name
Scroll Down
Users
UWT
All Users
Default
Default User
john
Public
desktop.ini
3D Objects
ansel
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
Helen_Contacts.pdf
Hobart Map.odt
IB_Data.odt
img20190814_16442778.jpg
img20190814_16450113.jpg
Initials.odt
Joystick Adjustment.odt
Label.odt
Lexi_Pictures.odt
Make an eBook How to Easily Create a PD
Malwarebytes_Fix.odt
Mr Polly.odt
My_Blender_Note.txt
Open_Source_Software.odt

Figure 3.8

3.6 Opening Files

When the File Browser Editor has been opened, replacing the 3D Viewport Editor, it is merely allows you to search your file system to locate folders and files or create a new folder. **You can not open files.**

Blender opens different types of files in different Editors (see Chapter 04 Objects in the 3D Viewport Editor). **My_Blender_Note.txt** is a text file therefore you would open it in the **Text Editor.** Blender files with the **.blend** extension open in **Blender** in the 3D Viewport Editor. Image files would open in the UV Image Editor.

To demonstrate, consider the file named **My_Blender_Note.txt.** The .txt suffix indicates that it is a text file, therefore, it is opened in the Text Editor. Replace the 3D Viewport Editor with the Text Editor. To open My_Blender_Note.txt click **Text** in the Header and select **Open** or click **Open** (Figure 3.9).

Figure 3.9

Clicking on either **Open** options displays the **Blender File View** panel which is essentially the Blender File Browser for the Text Editor. If you were in the Image Editor or the 3D Viewport Editor the Blender File View panel would be applicable to the Editor that was opened allowing you to select and open files applicable to the particular Editor.

Having navigated to the Documents Folder in the File View panel and scrolling you may be disappointed to find **My_Blender_Note.text** is nowhere to be found. Text files can be hidden? Click on the **Display button** in the upper RH corner of the panel and check **Text Files** in the display menu. Text Files will be shown, Select (click to highlight) and click Open Text.

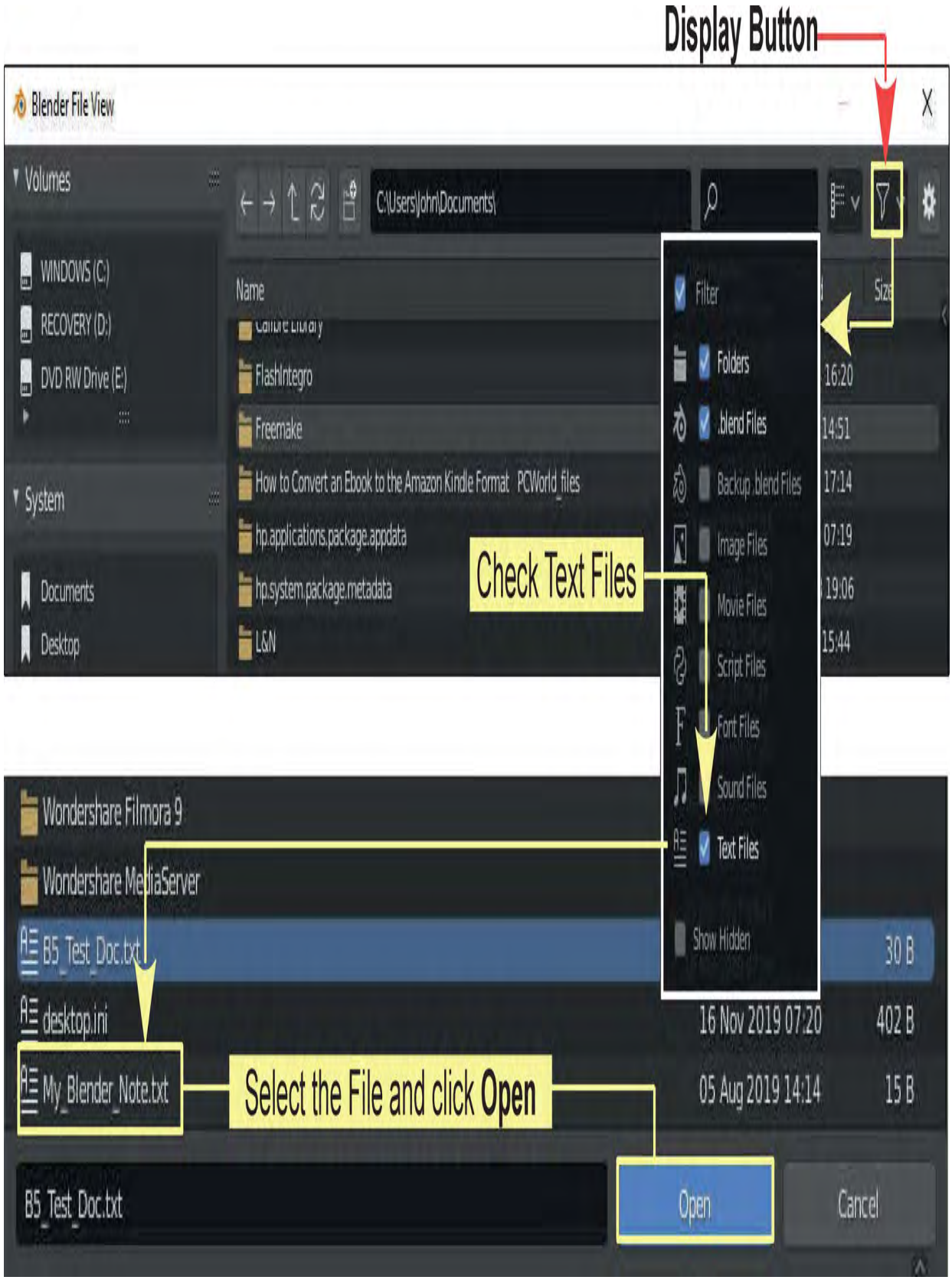

Display Button
Blender File View
Volumes
C:\Users\John\Documents\
WINDOWS (C:)
RECOVERY (D:)
DVD RW Drive (E:)
System
Documents
Desktop
Name
FlashIntegro
Freemake
How to Convert an Ebook to the Amazon Kindle Format PCWorld files
hp.applications.package.appdata
hp.system.package.metadata
L&N
Filter
Folders
.blend Files
Backup .blend Files
Image Files
Movie Files
Script Files
Font Files
Sound Files
Text Files
Show Hidden
Check Text Files
Wondershare Filmora 9
Wondershare MediaServer
B5_Test_Doc.txt
desktop.ini
My_Blender_Note.txt
16 Nov 2019 07:20
05 Aug 2019 14:14
30 B
402 B
15 B
Select the File and click Open
B5_Test_Doc.txt
Open
Cancel

Figure 3.10

3.7 File Browser Header Features

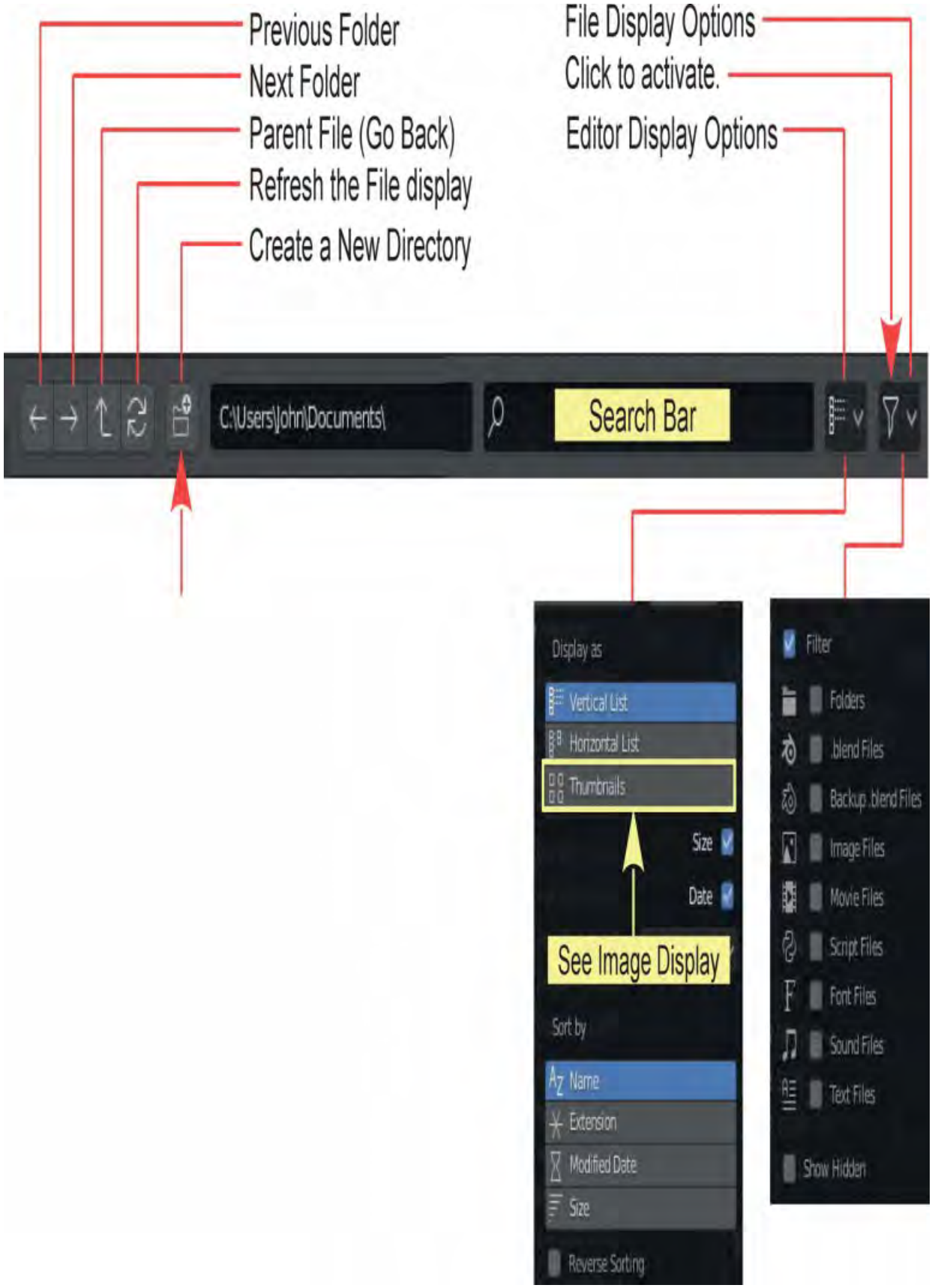

Previous Folder
Next Folder
Parent File (Go Back)
Refresh the File display
Create a New Directory
File Display Options
Click to activate.
Editor Display Options
C:\Users\John\Documents\
Search Bar
Display as
Vertical List
Horizontal List
Thumbnails
Size
Date
See Image Display
Sort by
Name
Extension
Modified Date
Size
Reverse Sorting
Filter
Folders
.blend Files
Backup .blend Files
Image Files
Movie Files
Script Files
Font Files
Sound Files
Text Files
Show Hidden

Figure 3.11

3.8 Make a New Folder

Clicking the **Create New Directory** button creates a New Folder in the Directory that is open, in this case the My Documents directory. The New Directory is named **New Folder** but is displayed with a **Typing Cursor active** allowing you to enter your own name.

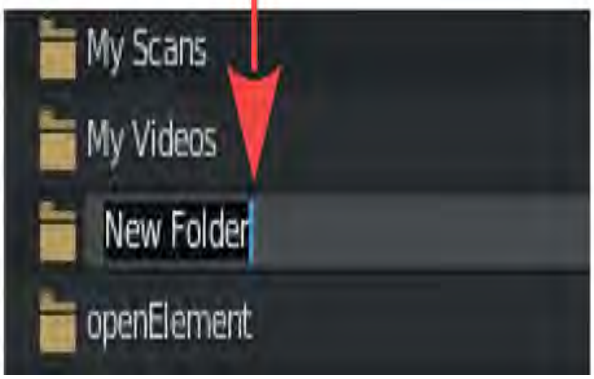


Figure 3.12

3.9 Display Options

The way in which Folders and Files are displayed can be changed in the **Editor Display Options.**

Figure 3.13

3.10 Saving Your Work

Besides being able to navigate the File Browser to find things that have been saved you will inevitably want to save your work. Its a good idea to be organised and create folders for different things.

Make a new folder in the C:\ Directory and name it **A_My_Blender_Stuff.**

Open the File Browser Editor.

Click on **Windows (C:)** under **Volumes** (upper **LH).** In the File browser Header click the Create New directory Button. New Folder is created. With the Typing Cursor active, backspace and retype **A_My_Blender_Stuff.** Press Enter.

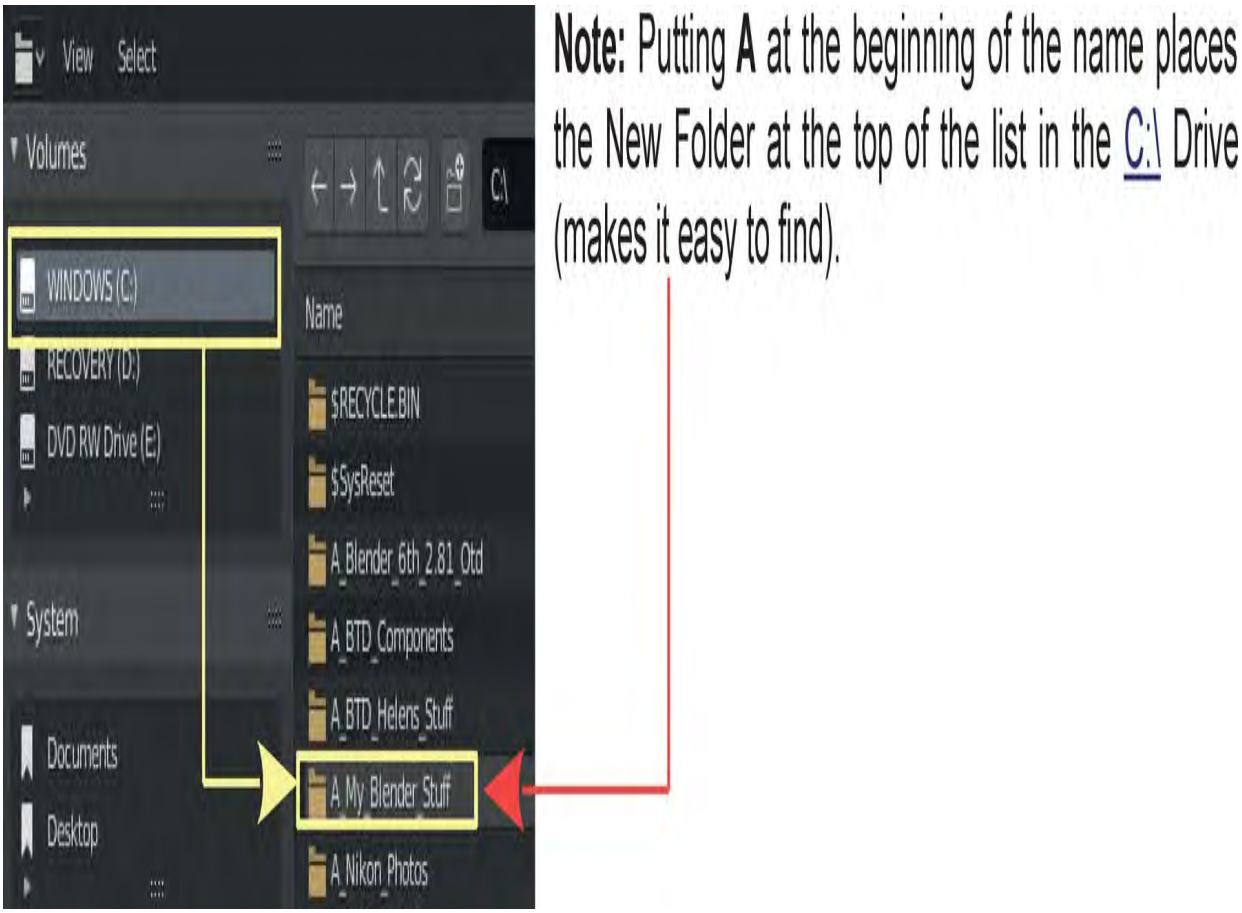


Figure 3.14

For the exercise a modified copy of the default Blender Scene will be saved to the new folder named **My_Blender_Stuff.**

With the new folder created, open the 3D Viewport Editor to display the default Blender Scene with the Cube Object. Replace the Cube with a Monkey Object. Adding the Monkey will distinguish the file to be saved from the original default Scene.

In the 3D Viewport Editor Header, click on **File** then click on **Save As** in the menu. The Blender File Viewer opens. Navigate to the Folder named **My_Blender_Stuff.** At the bottom of the File Viewer, where you see **untitled.blend,** click to display the Typing Cursor, backspace and retype a name for your new file, **My_New_File.blend** (If you forget the .blend suffix, Blender will obligingly add it for you). Click on **Save As** (highlighted blue).

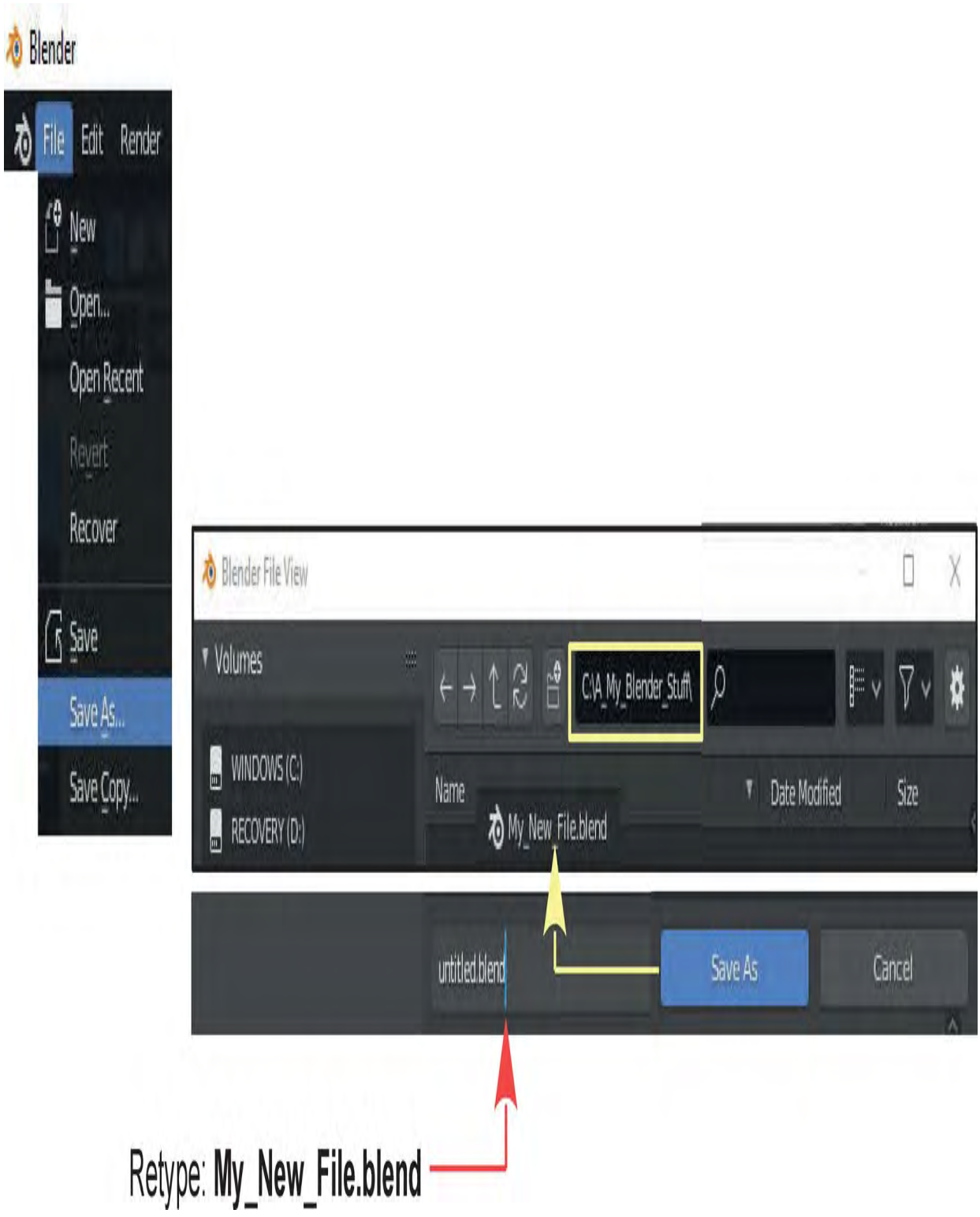


Figure 3.15

3.11 The Concept of Files

To save a file? What does this mean? Its easy to say, **save a file,** but what are you actually doing when you save? The chapter started by discussing saving a file created in Windows Notepad. This was a simple text file. A text file contains data which displays letters and words on your screen i.e. Text. An image file contains data which displays a picture. A music file plays music. Each file type uses a different application (App) or program to generate the display or, in the case of a music file, play the music. Sure! You know all this but the point is; a Blender file contains a combination of data organised into separate parts or elements.

When you save a Blender file you save all the elements.

To show you what this means in practical terms, open Blender with the default Scene. Click on **File** in the **Blender Screen Header** then click on **Append.** The **File Viewer** opens. Navigate to the file you previously saved named **My_New_File.blend** (Figure 3.16).

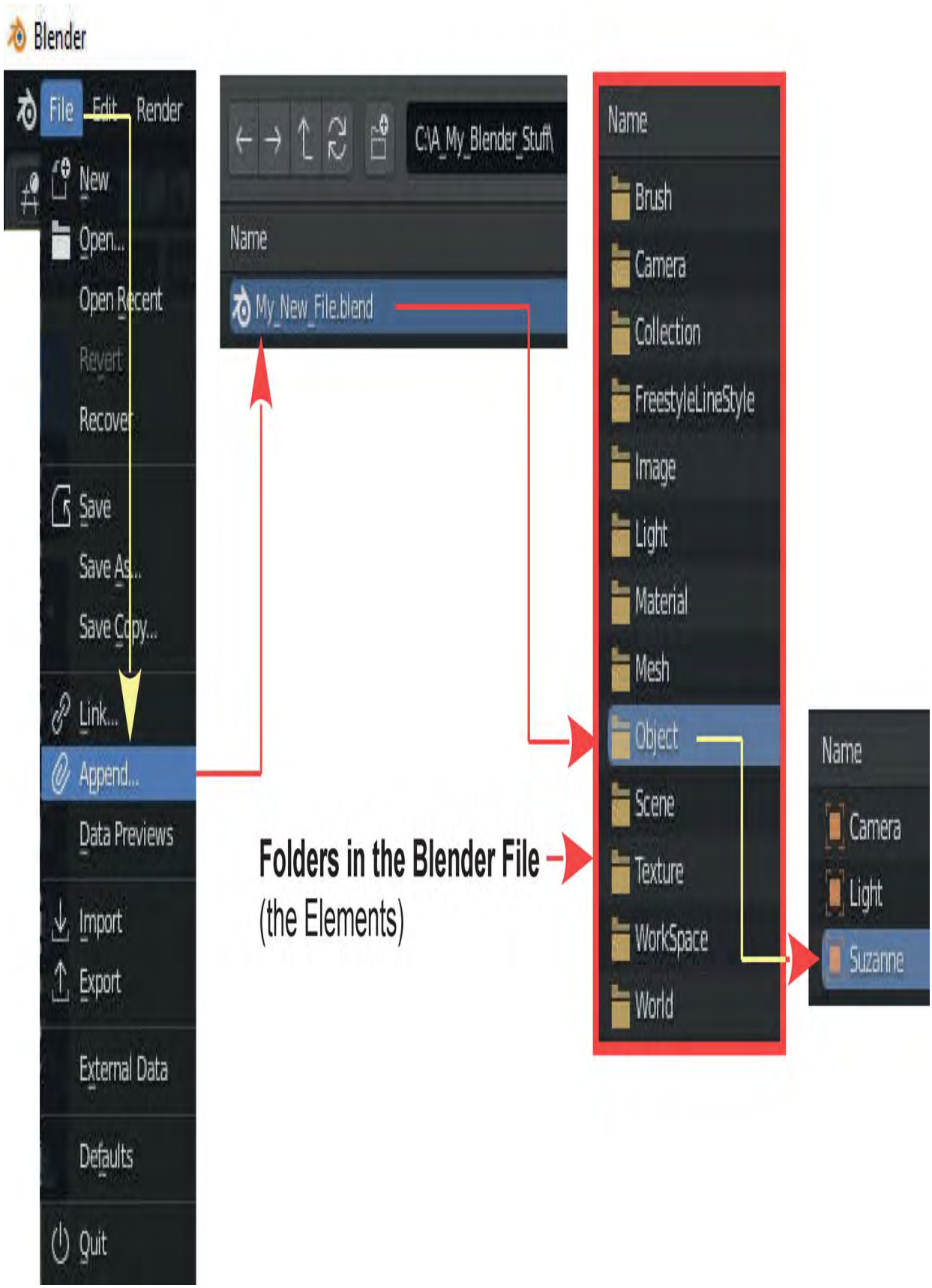

Blender
File
Edit
Render
New
Open...
Open Recent
Revert
Recover
Save
Save As...
Save Copy...
Link...
Append...
Data Previews
Import
Export
External Data
Defaults
Quit
C:\A_My_Blender_Stuff\
Name
My_New_File.blend
Name
Brush
Camera
Collection
FreestyleLineStyle
Image
Light
Material
Mesh
Object
Scene
Texture
WorkSpace
World
Name
Camera
Light
Suzanne
Folders in the Blender File
(the Elements)

Figure 3.16

3.12 The Append or Link Command

You can insert elements from one Blender (.blend) file into another Blender file. To do this you select the **Append** or **Link** commands from the File pull-down menu in the Blender Screen Header (Figure 3.16). An element could be a model you have created.

Append takes data from an existing file and adds it to the current file. **Link** allows you to use data from an existing file in the current file but the data remains in the existing file. In the latter case the data cannot be edited in the current file—if the data is changed in the existing file, the changes show in the current file the next time it is opened.

At this stage the foregoing is probably somewhat difficult to understand, therefore working through an example will help to clarify the meaning. You will have to jump the gun a little and follow some procedures without explanation. The detail of the procedures will be covered later but at the moment only be concerned with the file system navigation involved.

Selecting **Append** or **Link** opens the File Viewer allowing you to navigate to the Blender file you wish to select elements from. You can **Append** anything, including cameras, lights, meshes, materials, textures, scenes, and objects. By appending Objects, any materials, textures and animations that are linked to that object will automatically be imported with the Object. Clicking the LMB on an Object will select it. Pressing the A key will deselect.

To clarify this procedure start a new Blender file (open Blender) with the default Scene containing the Cube Object. Select the **Append** command as previously described and navigate to the file **My_New_File.blend**. Open the **Object directory** (the Objects folder, Figure 3.15). Click on the Object named **Suzanne** to highlight it, then click on the **Append** button in the lower RH corner of the Viewer (Figure 3.17).

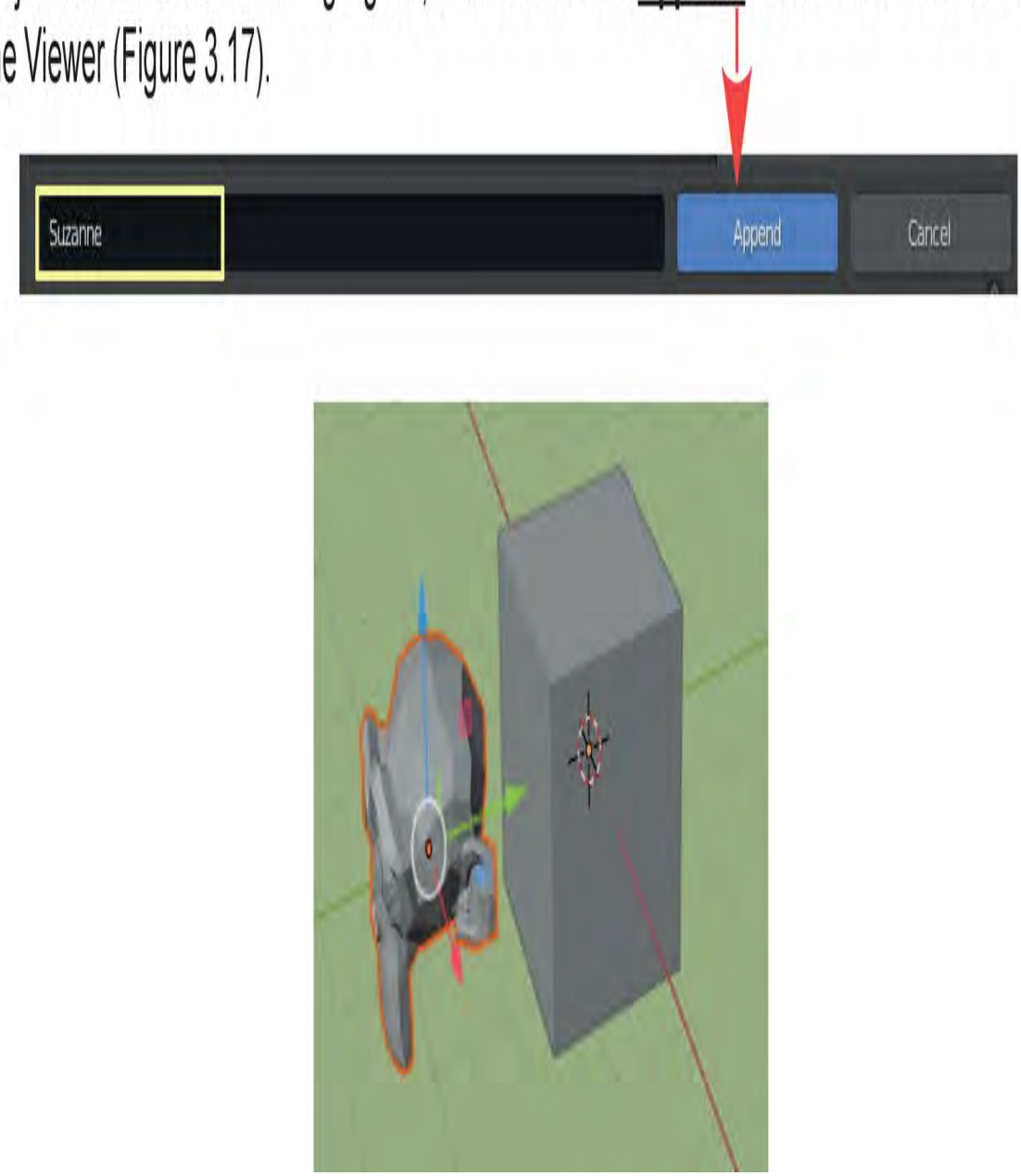


Figure 3.17

Suzanne Appended in the 3D Viewport Editor.

You can Append any Object from any Blender file.

Note: When Appended an Object is located in the Scene at its location in the the file from which it originated.

3.13 Importing Objects

The **Make Human** program is freely available. Other programs save files in one format but also give the option to export in another format. You will have to find the **Export** command in the program and match up the file type with one of the file types in Blender's import menu.

Note: The file type options shown in Figure 3.18 have been abbreviated as has the actual selection menu. **Collada (.dae)** is shown while **Pointcache (.pc2)** is not. To conserve space in the GUI, Blender has limited the file type display. Other file types are available as **Add-ons** in the **Preferences Editor** in the **Import-Export** category.

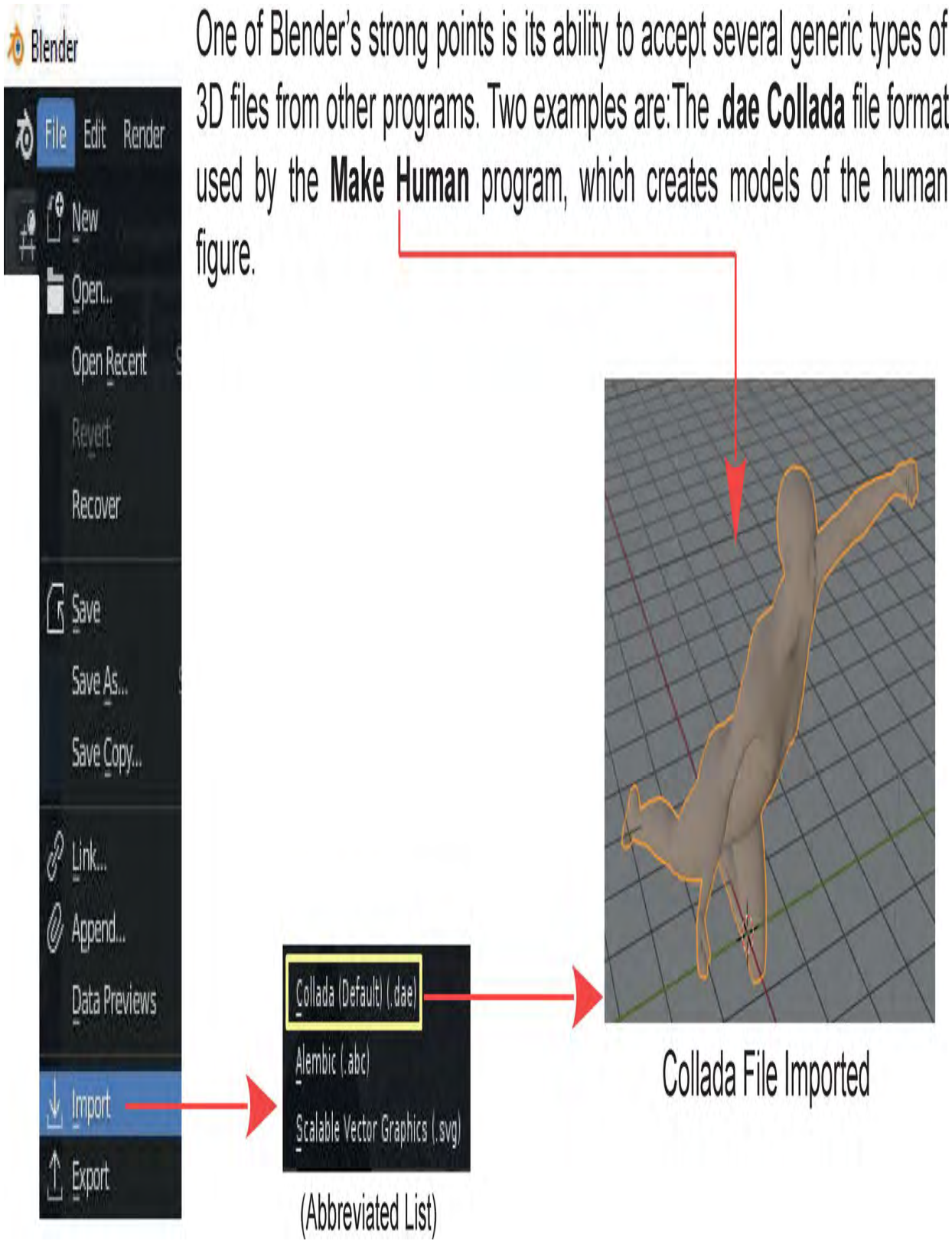

Blender
File Edit Render
New
Open...
Open Recent
Revert
Recover
Save
Save As...
Save Copy...
Link...
Append...
Data Previews
Import
Export
Collada (Default) (.dae)
Alembic (.abc)
Scalable Vector Graphics (.svg)
(Abbreviated List)
One of Blender's strong points is its ability to accept several generic types of 3D files from other programs. Two examples are:The .dae Collada file format used by the Make Human program, which creates models of the human figure.
Collada File Imported

Figure 3.18

3.14 Activating Import File Types

To import a **DXF file** (Autocad DFX Format-.dfx) into a Blender scene, open the **Preferences Editor** and click on **Add-ons** at the LHS of the Editor. In the list to the right select an **Import- Export Add-on.** Find the **File Type** you require and check (tick) the box adjacent to the name. The checked file type will display in white text and be available in the **Import** selection menu.

Note: When importing Blender files into other Blender files, remember to use the **Append** command instead of Import. In the **Append** command, select the file, then select what you would like to bring into the current file. You will usually use the Objects option.

The Preferences Editor – Add-ons – Import-Export

Note: You may type **Import** in the Search Bar to find Import Add-ons.

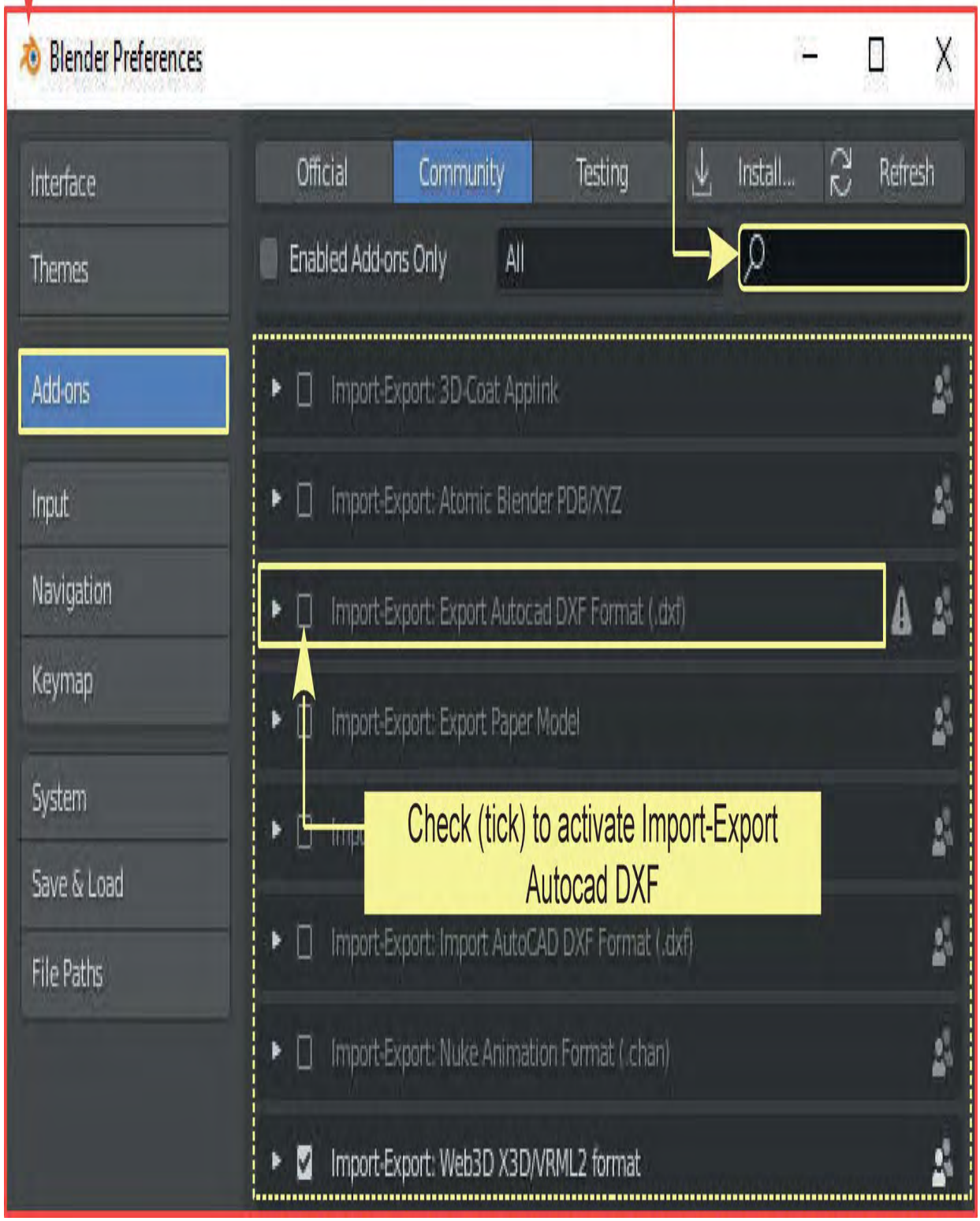


Figure 3.19

Note: Although Autocad DXF is titled Import-Export, activating the Add-on only provides an Import Function. You can not export from Blender to Autocad.

Clicking the file type opens the File Browser Editor where you navigate to the file containing the Object you wish to import. As an example, a model of a human figure has been created using the **Make Human Program.** The model has been exported as a **Wavefront.obj** file. You will see that this type of file is included in the default Import file list, **Wavefront OBJ.** Click on this file type to display the **File Browser Editor.**

Important: When you export a model from another program, know where it is saved on your computer and what the file name is.

The model exported from the **Make Human** program is named **Running_Man.obj.** The model was exported to:

Blender File View
Volumes
C:\A_My_Blender_Stuff\
WINDOWS (C:)
RECOVERY (D:)
DVD RW Drive (E:)
Name
Date Modified
Size
Operator Presets
Running_Man.obj
Include
Import OBJ
Cancel

Figure 3.20

Figure 3.21

In the File Browser Editor navigate and find your file. Click on the file name to select it (highlight) then click on **Import OBJ** in the lower RH corner

The Object Running_Man (Figure 3.21) is entered into the Blender 3D View Editor. **Note:** You will probably have to scale the model to suit.

3.15 Packing Data

If you plan to open a Blender file on other computers, you will need to select the **Pack All into .blend** option in the **File** menu under **External Data.** Textures and sounds are not automatically included in your Blender file in order to keep the file size down. Every time your file opens, the textures and sounds are placed into your file. If the files can't be found, you won't have any textures and sounds. If you **Pack the Data,** those files are included with the .blend file so they can be opened anywhere. Remember, your file size may become very large. You can also unpack data to bring the file size back down.

You may alternatively check **Automatically Pack Into .blend.**

4 Objects in the 3D View Editor

4.1 Modeling Workflow Philosophy
4.2 Starting a New File
4.3 Modifying the Scene
4.4 Object Mode and Edit Mode
4.5 3D View Editor Cursor
4.6 Selecting/Deselecting Objects
4.7 Adding Objects
4.8 Object Primitives
4.9 Locating the Editor Cursor
4.10 Deleting Objects
4.11 Duplicating Objects
4.12 Object Mode Manipulation
4.13 The Last Operator Panel
4.14 Tool Panel - Widgets
4.15 Manipulation Units
4.16 Measuring - Ruler/Protractor
4.17 Precision Manipulation
4.18 Coloring Objects
4.19 Other Types of Objects
4.20 Naming Objects
4.21 the Header Buttons Menu
4.22 Meta Shapes

The overview of Blender introduces the Editors in the default Screen arrangement and demonstrates how they interrelate. The fundamental concept in using Blender is to create a model, place the model in a Scene and perhaps animate the model to move creating an animation sequence. In the creation process the models and components of the Scene may be colored and textured and the Scene illuminated for effect.

To model an object or character, or a component of a Scene such as a landscape, a simple shape is placed in the 3D Viewport Editor and modified (modeled). The simple shape is called a **Primitive**. Blender has ten primitives from which to choose, which one you select will depend on what you want to create.

Instruction will be given with the default Screen Arrangement (GUI) with the **Eevee Render Engine** active.

Bear in mind that you have the options to use the **Cycles** or **Workbench** Render Engines. Workbench provides a simplified working environments for modeling when Scenes become complicated. Cycles allows you to work in a viewport that continually renders a **photo-realistic** view.

4.1 Modeling Workflow Philosophy

Blender has been introduced by studying the Graphical User Interface (GUI) with its arrangement of Editors and panels and having Keyboard and Mouse input commands explained. Knowing how to work the interface allows you to create something by Modeling.

Before you begin to Model you should understand the philosophy of the Blender process.

When you open Blender you are presented with the default 3D Viewport Editor showing a Scene containing a Cube Object. The Scene also has a Light and a Camera in place. To create something new, you start by saving this arrangement with a **new file name**. The new file is a starting point for

developing a new Scene with new Models. You modify the default Scene in the new file to whatever you require. Modifying a Scene will involve such things as moving and repositioning objects, reshaping objects, adding new objects, applying color, arranging lighting effects, positioning the camera etc.

4.2 Starting a New File

When beginning a new project it is advisable to start a new file with a new name. Start Blender and before changing anything in the default Scene click **File – Save As** and save the file in your **Blender_Stuff** folder with a meaningful name. Write down the name. You can save your work wherever you like as long as you remember what you named the file and where you saved it. Be familiar with saving and creating files and folders. Go back and review Chapter 3 if necessary.

> **Note:** Depending on the version of Blender you have, the program may not prompt you to save your file when closing . Remember to always save your work often and don't forget the .blend suffix!

After saving the default Scene as a new file, the new file is open in the Screen ready to be modified. The new Blender file will display the default arrangement of Editors and the 3D Viewport Editor will display the default Scene. If you have closed Blender after saving you will have to reopen the new file.

When you restart Blender the default Screen arrangement is displayed. To reopen the file that you saved, click on **File** in the **Blender Screen Header** and click on **Open** in the **File** menu. The **File Viewer** opens (Figure 4.1) where you navigate to where you saved your file. Click on your file name then click **Open** in the lower RH corner of the Editor

Figure 4.1

4.3 Modifying the Scene

Any changes you make to the Scene will be construed as modifying the Scene. For example, changing the shape of an Object is a modification. Changing the shape of an Object is called modeling. Another basic modification would be to move an Object in the Scene. Another Scene modification is to add an Object. There are many modifications that can be made.

4.4 Object Mode and Edit Mode

In the **3D Viewport Editor** there are two basic display modes, **Object Mode** and **Edit Mode**. With an Object selected in the 3D Viewport in Object Mode you Translate (move), Rotate and Scale the Object. In Edit Mode you change the shape of the Object. Note: This is a simplistic description of the

operations, there is more to it than that. Look at the Cube Object in the default Scene.

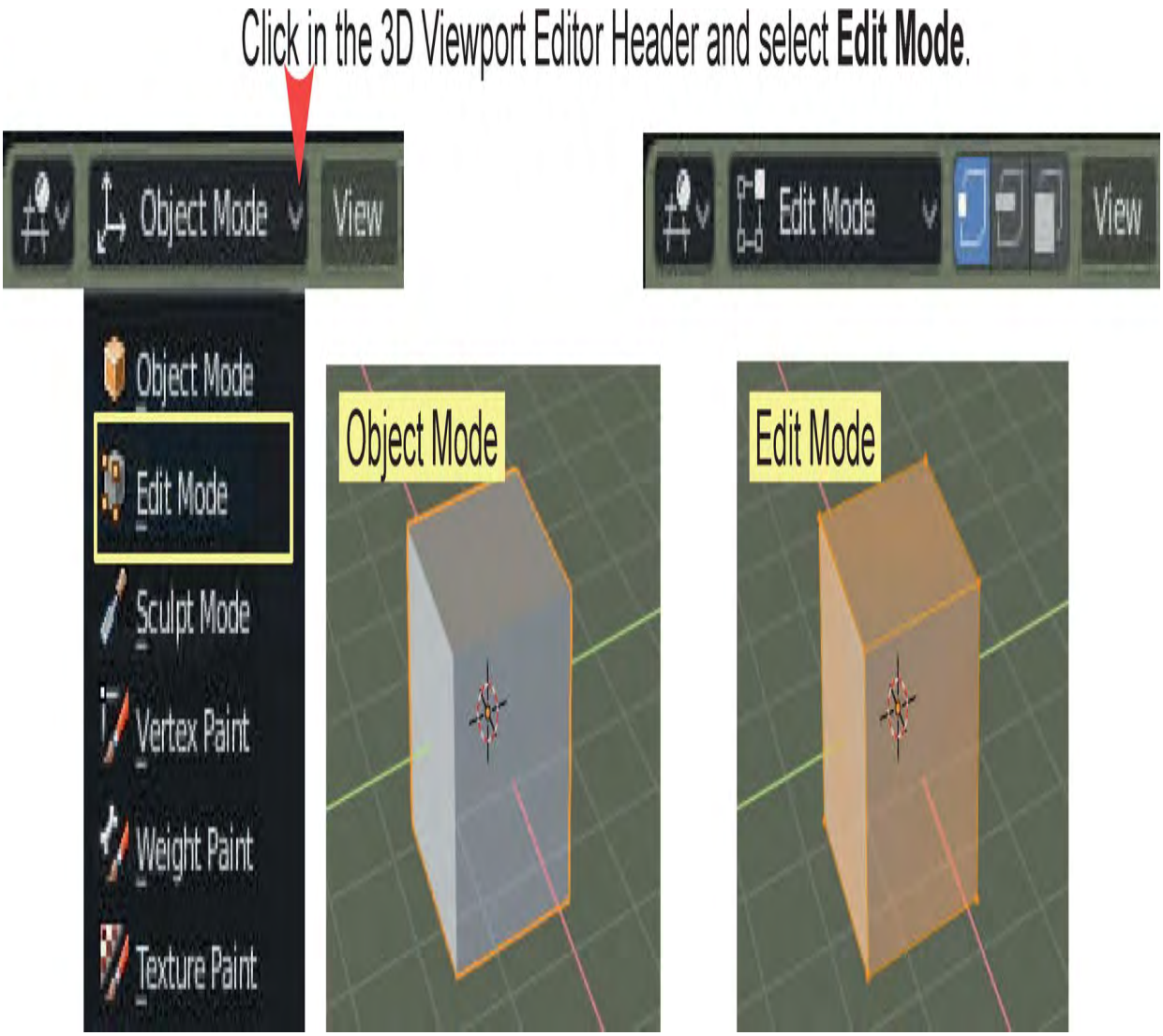


Figure 4.2

Alternative: Press the **Tab Key** on the Keyboard to toggle between modes.

In **Object Mode** the Cube displays as a solid gray Object with an orange outline. The orange outline indicates that the Cube is **selected**. Being selected means that the Cube may be manipulated in the Scene (Translated (Moved), Rotated or Scaled).

In **Edit Mode**, when the Cube is selected, it is shown with an orange tint with orange edges and with little orange dots at each corner. The orange dots are called **Vertices**, the orange edges are **Edges** while the orange tinted surfaces are **Faces**. In Edit Mode you select either of these elements individually and manipulate them to change (model) the shape (see Chapter 5).

4.5 3D Viewport Cursor

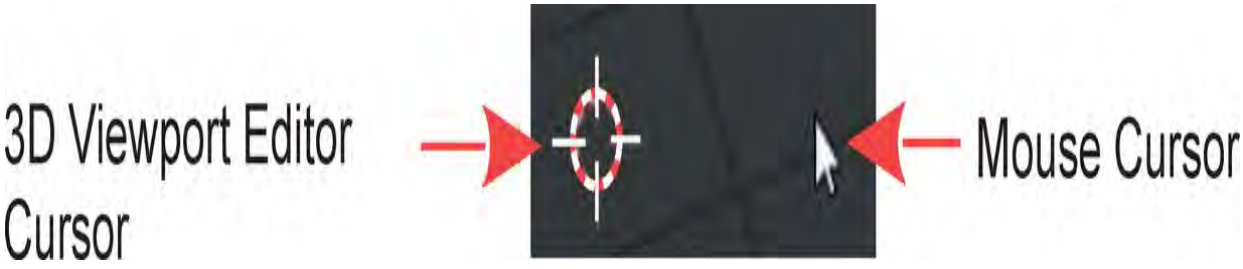


Figure 4.3

When working in the 3D Viewport Editor there are two Cursors; the **Mouse Cursor** and the **Editor Cursor**. By default the **Editor Cursor** is located at the center of the Scene. The center of the default Cube Object is also at the center of the Scene. Objects may be positioned by using the Editor Cursor. To understand positioning using the Editor Cursor you must first understand selecting, deselecting and adding Objects.

4.6 Selecting – Deselecting Objects

In the default Scene the **Cube Object is selected as shown by the orange outline.** To deselect the Cube **LMB click** (left mouse button) an empty space in the Editor (the orange outline cancels). To **select** the Cube again, position the Mouse Cursor on the Cube (mouse over) and

click **LMB** .

Note: If you deselect then press the **A Key,** all Objects in the Scene will be selected. If you inadvertently select all Objects click an empty space to deselect all Objects. Click LMB on a Object to select.

B Key Selection

An alternative selection method is to press the **B Key** on the Keyboard with the Mouse Cursor in the 3D Viewport Editor panel. In this case, pressing the B Key displays cross hair lines in the Editor panel which you drag, forming a rectangle, around an Object **or multiple Objects**. Release the mouse button to select (orange outline appears).

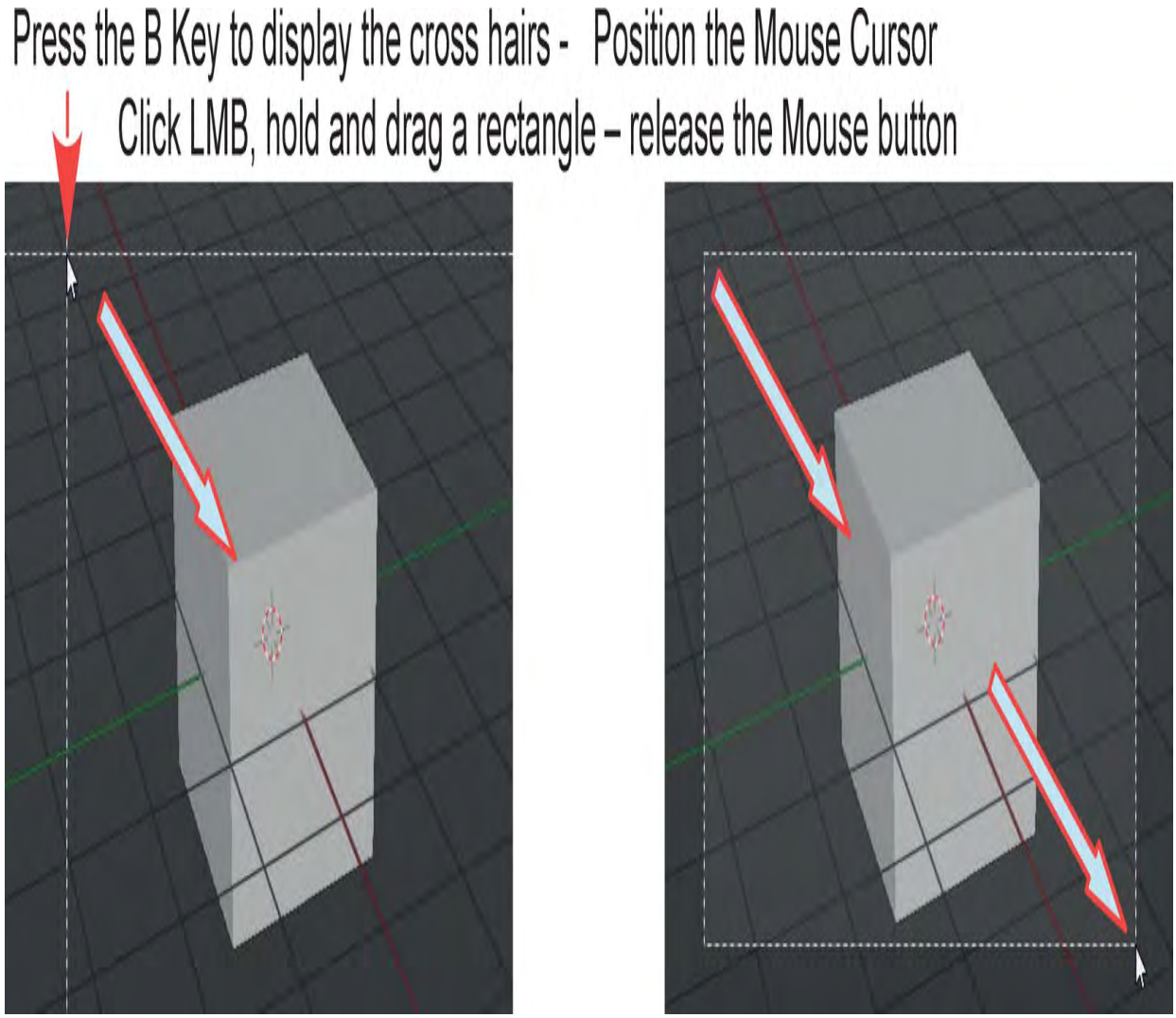
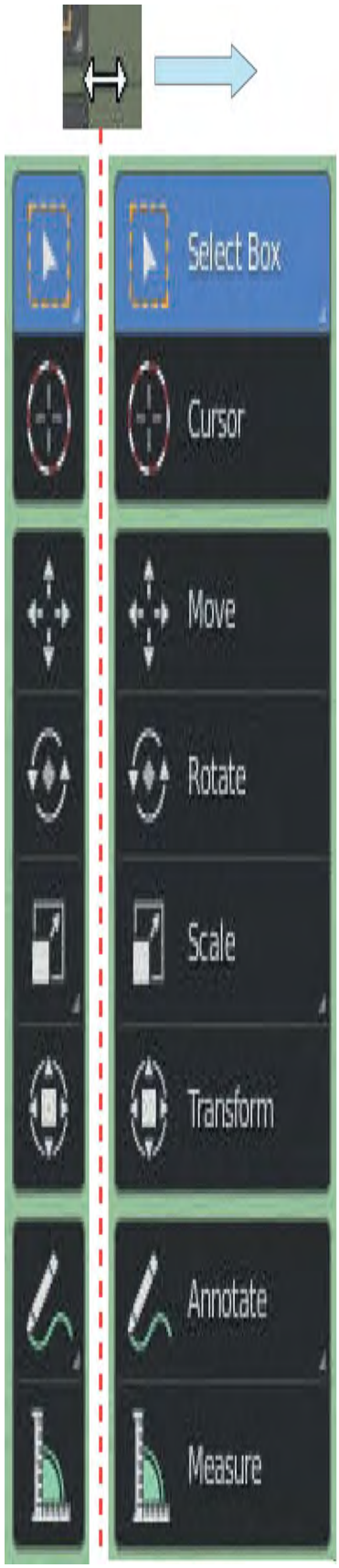


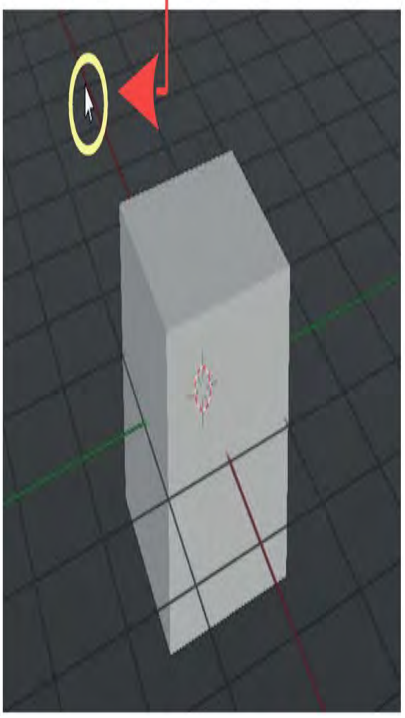
Figure 4.4

Note: The default **Tool Panel** is minimised. Mouse over, click, hold and drag the double headed arrow to expand and display the names.

Another method of selection is to use the **Border Select Tool** in the **Tool Panel** at the LH side of the 3D Viewport Editor. In this case, click the **Select Box** button, position the Mouse Cursor, click LMB, hold drag the mouse, drawing a rectangle around the Object or Objects to be selected. Release the Mouse button to select (as previously described).



Click the **Select Box** button, position the **Mouse Cursor.**



Click, Hold and Drag a Rectangle

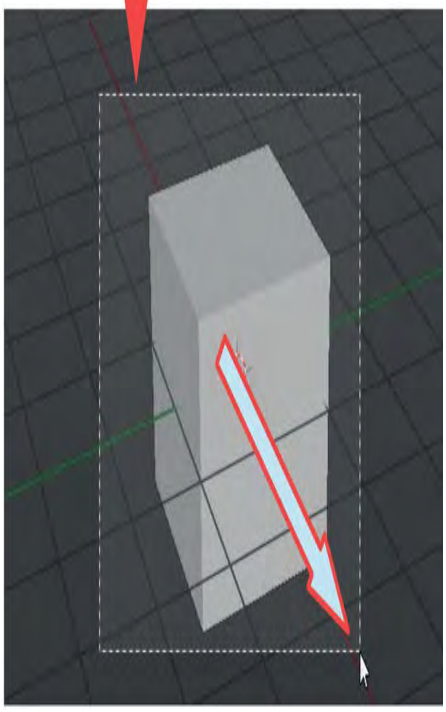


Figure 4.5

C Key Selection

Yet another method of selection is to position the Mouse Cursor in the 3D Viewport Editor and press the **C Key**. In this case the Mouse Cursor becomes a **Selection Circle** which you position over the Object or Objects to be selected. When first displayed the circle is relatively small.

Unless the circle encapsulates a whole Object the Object will not be selected. To increase or decrease the size of the circle **Scroll MMB**. With the circle surrounding the Object click LMB to select. Press **Esc** to cancel the selection circle.

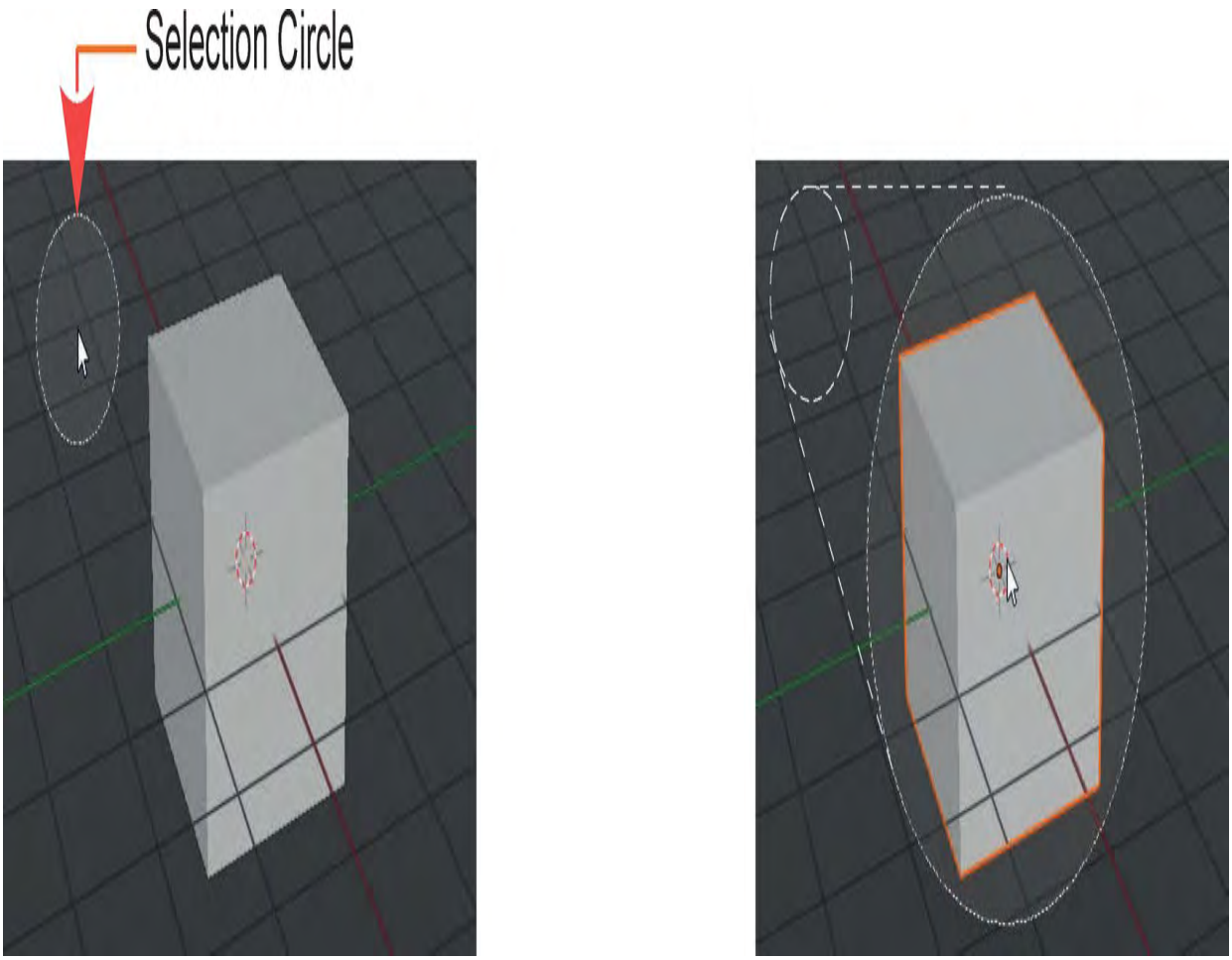


Figure 4.6

Scroll MMB to increase the size of the Circle.

Position over the Cube and click LMB to select. Press **Esc** to cancel the Circle.

4.7 Adding Objects

The default Blender Scene contains three Objects; a Cube, a Camera and a Light. The Camera and Light are special Objects which perform functions but do not **render** as part of an image or animation (see Chapter 15). The Cube, on the other hand, does render. The Cube is one of ten Objects called **Primitives**, which are the starting point for modeling. You modify the shape of a primitive into a model of a character or as part of the Scene background.

4.8 Object Primitives

Figure 4.7 shows the ten default **Primitives** available for selection. They have been colored to distinguish them. When added to a Scene they are entered with the default gray color. A **Primitive** shape entered into the scene is referred to as an **Object**. Blender automatically names Objects according to the shape i.e. **Cube**, **Sphere**, **Cone** etc. When you reshape (modify) the primitive Object to make a model you will rename it. Primitives are entered in the 3D Viewport Editor with the Editor in **Object Mode.**

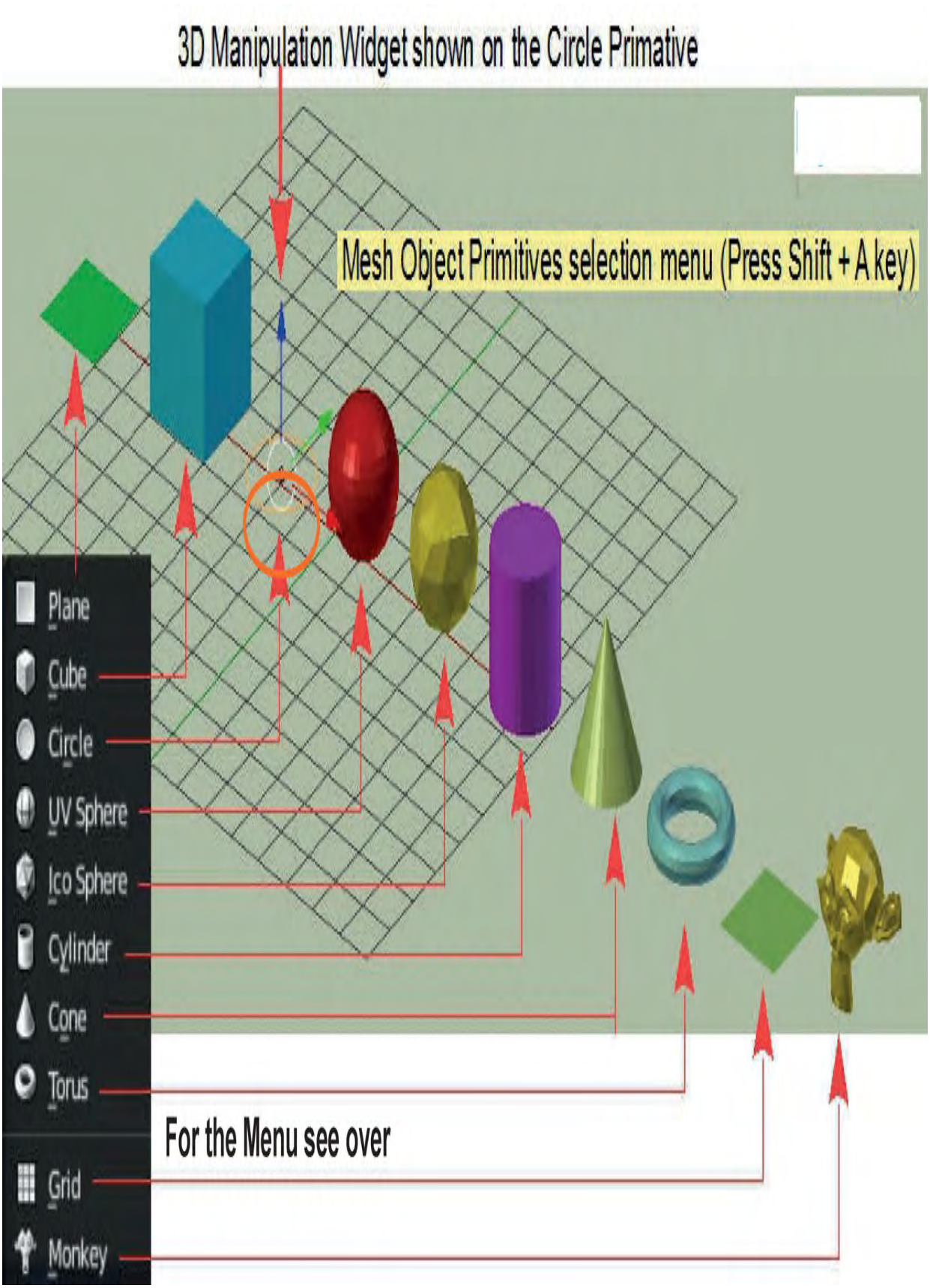

3D Manipulation Widget shown on the Circle Primative
Mesh Object Primitives selection menu (Press Shift + A key)
Plane
Cube
Circle
UV Sphere
Ico Sphere
Cylinder
Cone
Torus
For the Menu see over
Grid
Monkey

Figure 4.7

To add a new Object into a Scene you click on **Add** in the **3D Viewport Editor Header** then click **Mesh** to add a new Object. This displays the **Add Mesh menu**.

An alternative way to display the **Add** menu is to press **Shift + A Key**.

In either case, with the Add menu displayed you click **Mesh** to display the selection menu where you click to select one of the **Primitives**. Selecting a Primitive enters it in the Scene at the location of the **3D Viewport Editor Cursor**.

4.9 Locating the 3D Viewport Editor Cursor

By default the **3D Viewport Editor Cursor** is located at the center of the Scene which is the center of the 3D World. Also by default, the Cube Object is located at the same point.

Remember; Objects added (entered) in the Scene are located at the position of the **3D Viewport Editor Cursor**, therefore, you may wish to relocate the Cursor to position Objects.

Hold Shift and **click RMB** to position the Cursor. Alternatively activate the **Cursor Tool** in the Tool Panel. With the Tool activated you can click **LMB** anywhere in the 3D Viewport Editor to position the Cursor. With the Tool activated you can click LMB to relocate the Cursor and click hold and drag the Cursor to a new location.

Don't forget to deactivate the Cursor Tool so you can LMB click to select and deselect Objects. To deactivate the Cursor Tool click on the **Select Box** Tool (uppermost in the panel).

The Cursor Pie Menu (Snap Tool)

Yet another control for locating the Cursor and positing Objects is the **Snap Tool Pie Menu**. With an Object selected, press **Shift + S Key** to display the

Pie Menu

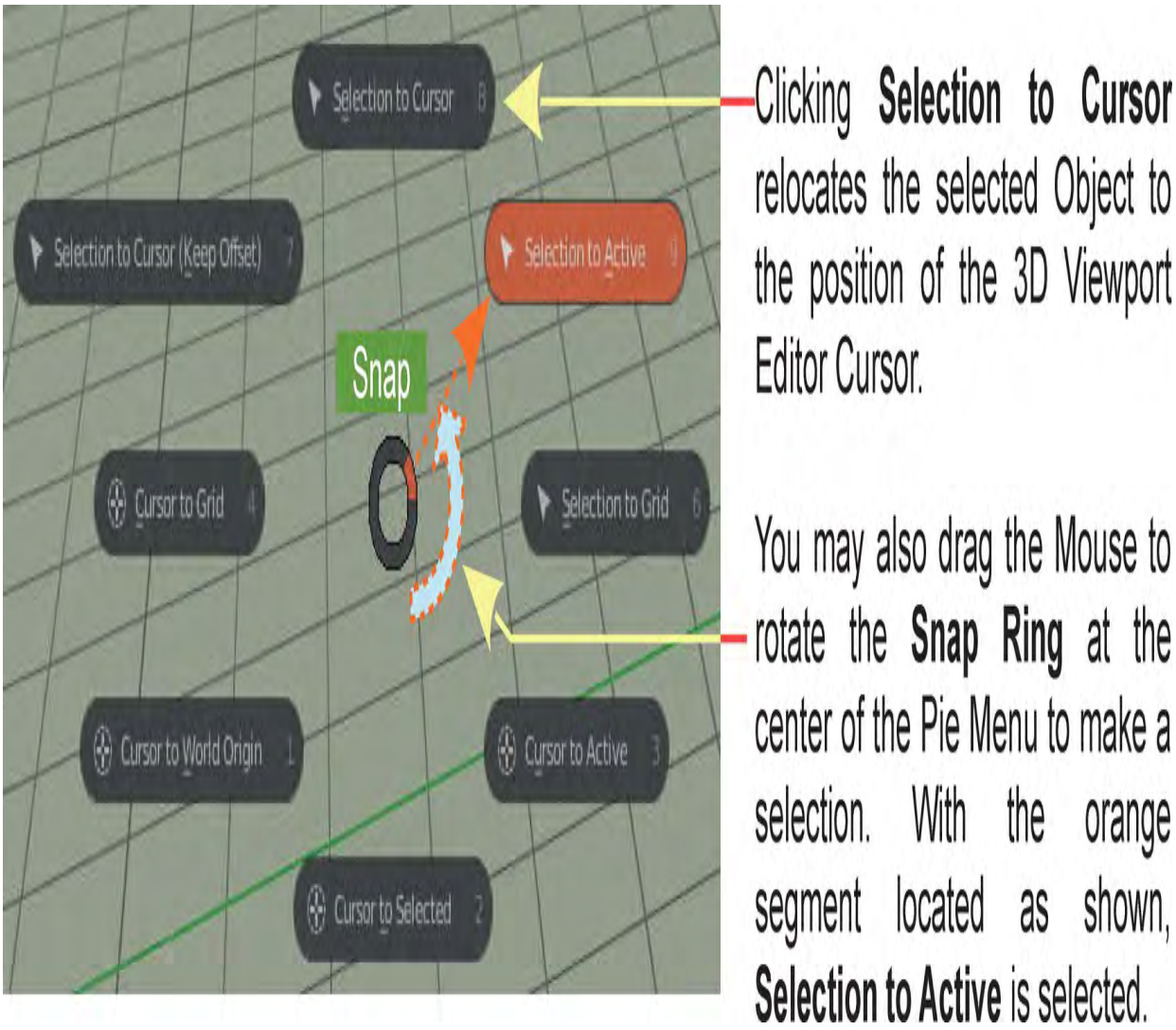


Figure 4.8

4.10 Deleting Objects

To Delete (remove) an Object from the Scene, select the Object then press the **X Key**. The **OK Delete** Click **Delete** to delete or **Esc** to cancel.

Figure 4.9

Note: You may select multiple Objects for deletion.

4.11 Duplicating Objects

When you want to **duplicate an Object** (make an identical copy), select the Object and press **Shift + D Key**. A new Object is created occupying the same space as the original. The new Object is in Move Mode (able to be moved by dragging the Mouse) as indicated by a white outline. To reposition in the Scene, drag the Mouse and click LMB when in place.

4.12 Object Mode Manipulation

The three basic manipulation controls are: **Translate, Rotate and Scale**.

Translate: To move an Object freely in the plane of the view, press the **G Key** (Grab Mode) with the object selected and drag the Mouse. In Grab mode the outline turns white. To lock the movement to a particular axis, press the G Key + X, Y, or Z. G key + Y restricts the movement to the Y (green) Axis (Figure 4.10).

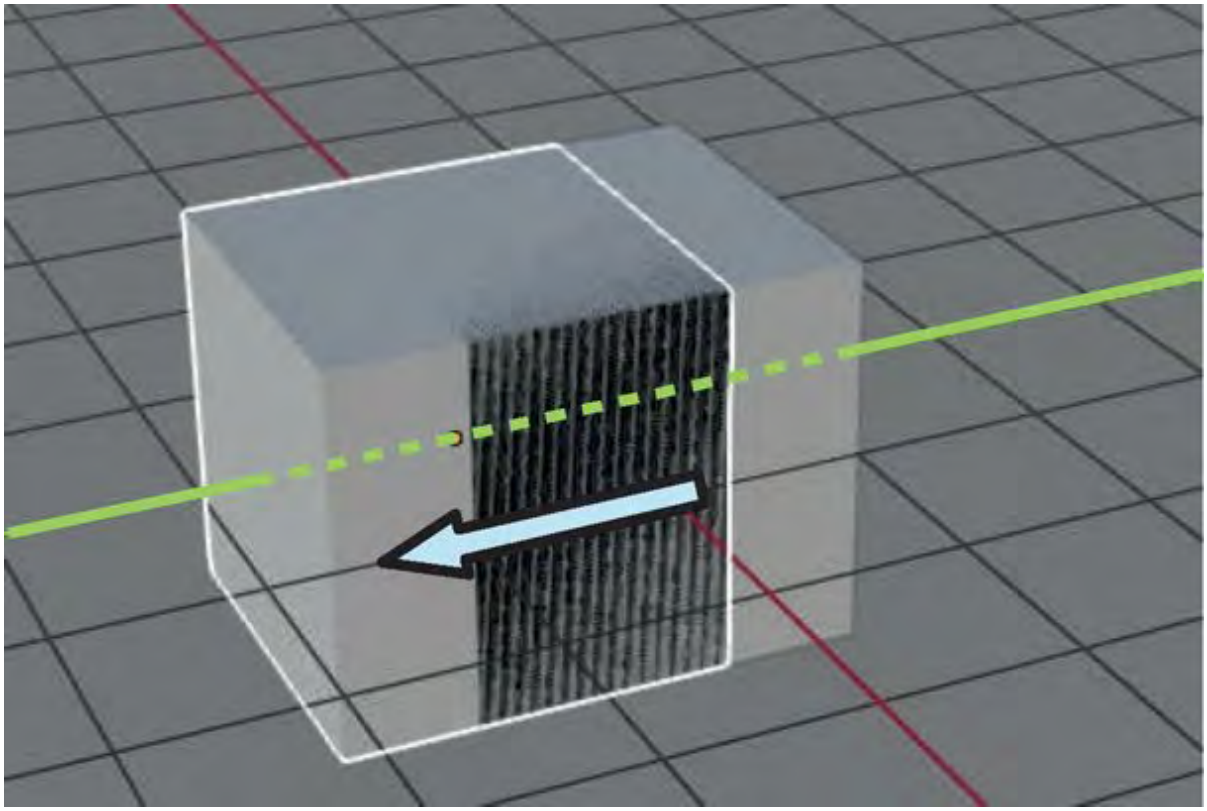


Figure 4.10

Scaling: To scale an Object (make larger or smaller), press the **S Key** and drag the mouse. To lock the scale to a particular axis, press S Key + X, Y, or Z. To scale by a specific value press S Key + Number Key (S + 2 + Y = Scale twice on the Y Axis (Figure 4.11)

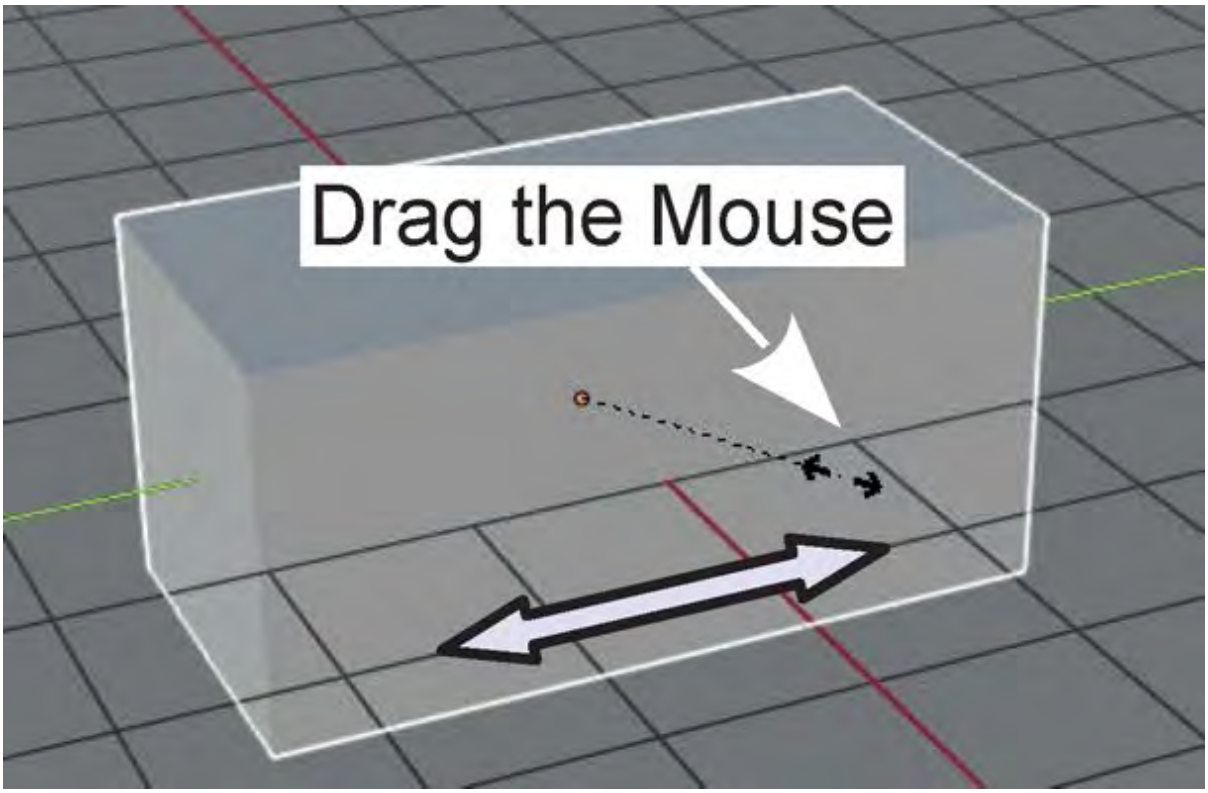


Figure 4.11

Rotating: To rotate an Object, press the **R Key** and move the mouse about the **Object's center**. To lock the rotation to an axis, press the R Key + X, Y, or Z. To rotate a set number of degrees, press R + the number of degrees of rotation and press Enter. R+30 rotates the Object 30 degrees. R + Y + 30 rotates the Object 30 degrees about the Y- Axis (Figure 4.12).

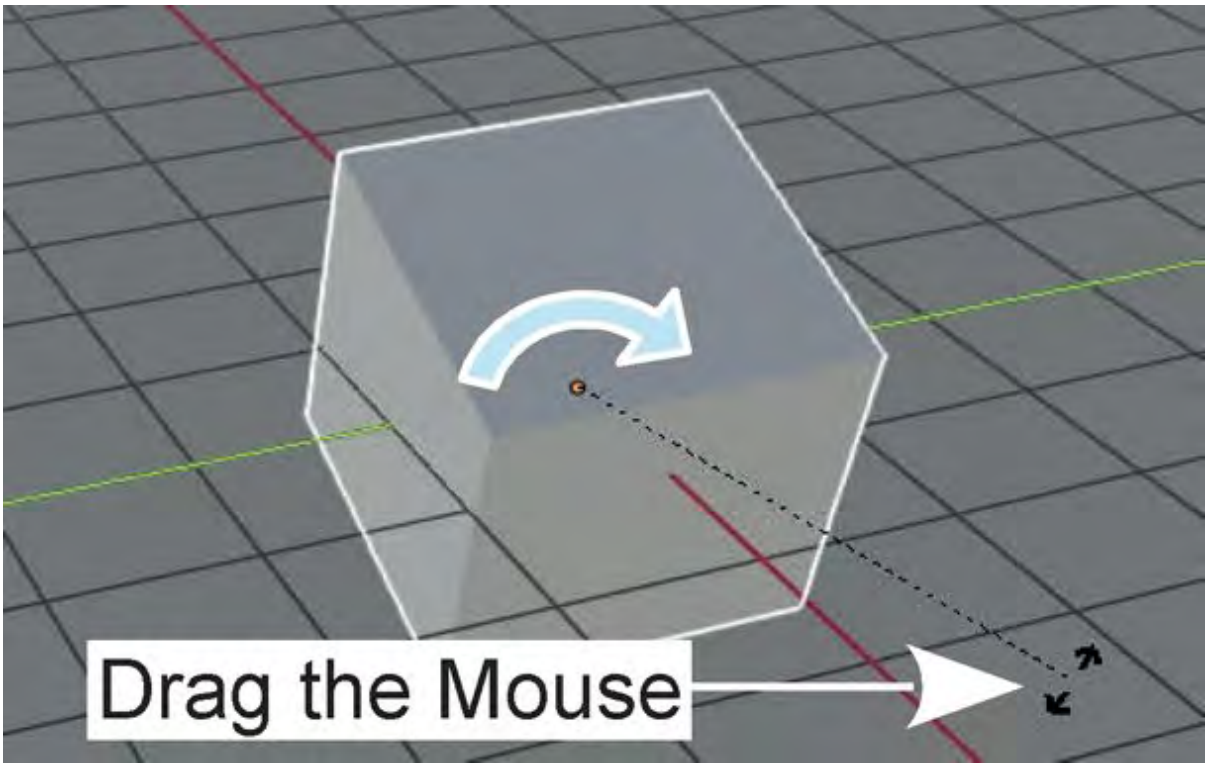


Figure 4.12

4.13 The Last Operator Panel

When an Object is manipulated a panel displays at the lower left hand side of the 3D Viewport Editor. This panel shows information relating to the last operation performed in the Editor. The panel also allows values to be adjusted, providing a means of fine tuning or correcting the operation.

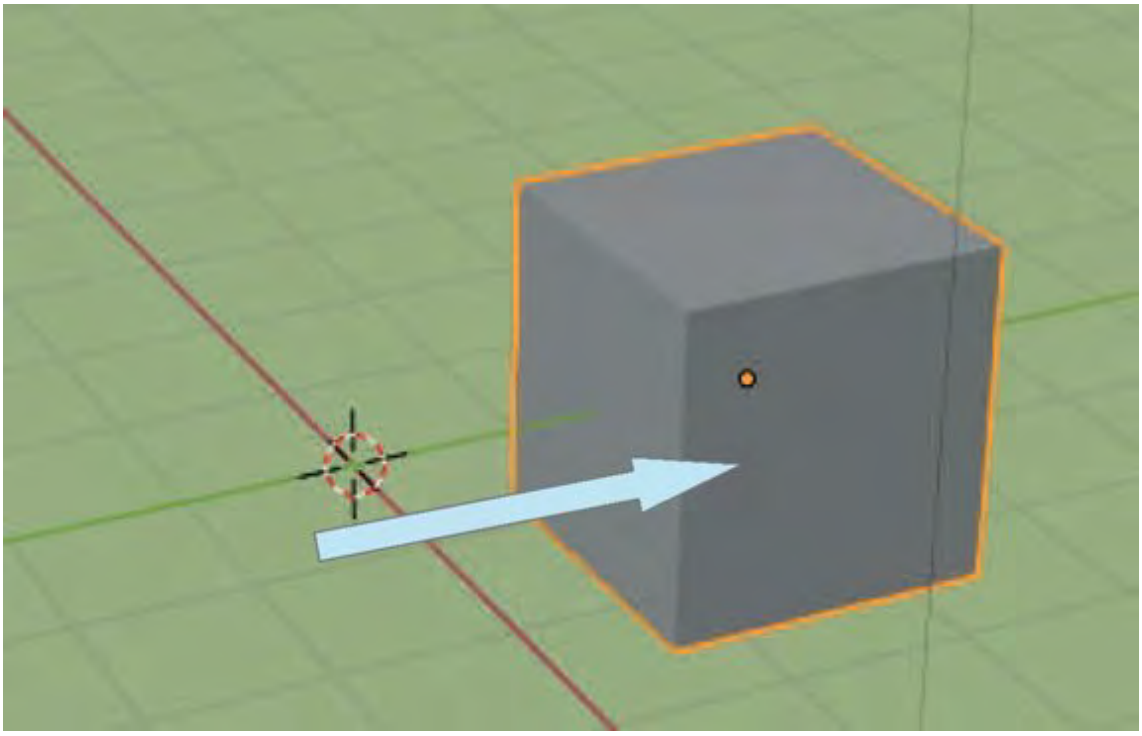


Figure 4.13

Consider Translation of the default Cube Object along the Y Axis. When the Cube is moved a panel displays in the lower left of the 3D Viewport. The name of the action is shown in the panel, in this case **Move**.

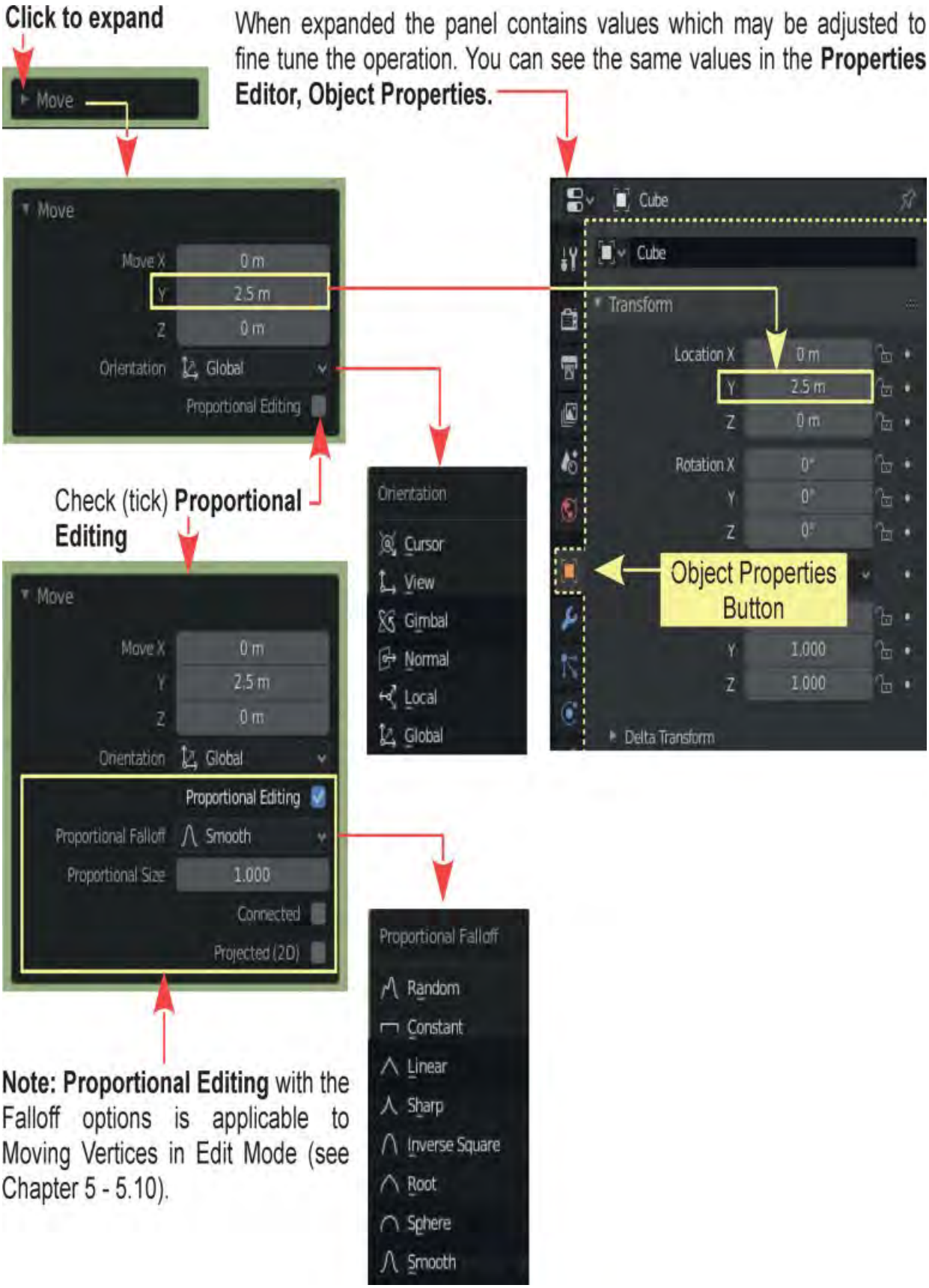

Click to expand
Move
When expanded the panel contains values which may be adjusted to fine tune the operation. You can see the same values in the **Properties Editor, Object Properties.**
Move
Move X 0 m
Y 2.5 m
Z 0 m
Orientation Global
Proportional Editing
Cube
Cube
Transform
Location X 0 m
Y 2.5 m
Z 0 m
Rotation X 0°
Y 0°
Z 0°
Object Properties Button
Y 1.000
Z 1.000
Delta Transform
Orientation
Cursor
View
Gimbal
Normal
Local
Global
Check (tick) **Proportional Editing**
Move
Move X 0 m
Y 2.5 m
Z 0 m
Orientation Global
Proportional Editing
Proportional Falloff Smooth
Proportional Size 1.000
Connected
Projected (2D)
Proportional Falloff
Random
Constant
Linear
Sharp
Inverse Square
Root
Sphere
Smooth
Note: Proportional Editing with the Falloff options is applicable to Moving Vertices in Edit Mode (see Chapter 5 - 5.10).

Figure 4.14

When values are altered in the Last Operator panel the corresponding value in the Properties Editor is also adjusted.

Note: The **Move Y** value in the Last Operator panel shows the distance moved from the original location. The Location Y value in the Properties Editor shows the actual location on the Y Axis of the Scene. Moving the Cube a further 1m on the Y Axis will show 1m in the Last Operator panel (distance moved) and 3.5m in the Properties Editor (the Location).

4.14 Tool Panel –

Expand the Tool Panel to display the Tool Names (see 4.6).

In the **3D Viewport Editor, Tool Panel** (Oject Mode) at the left hand side you will see **Transform, Move, Rotate** and **Scale** buttons. Clicking on a button (highlights blue) displays a **Manipulation Widget** at the center of the selected Object. To use a Widget click LMB on a red, green or blue handle, hold the mouse button and the mouse.

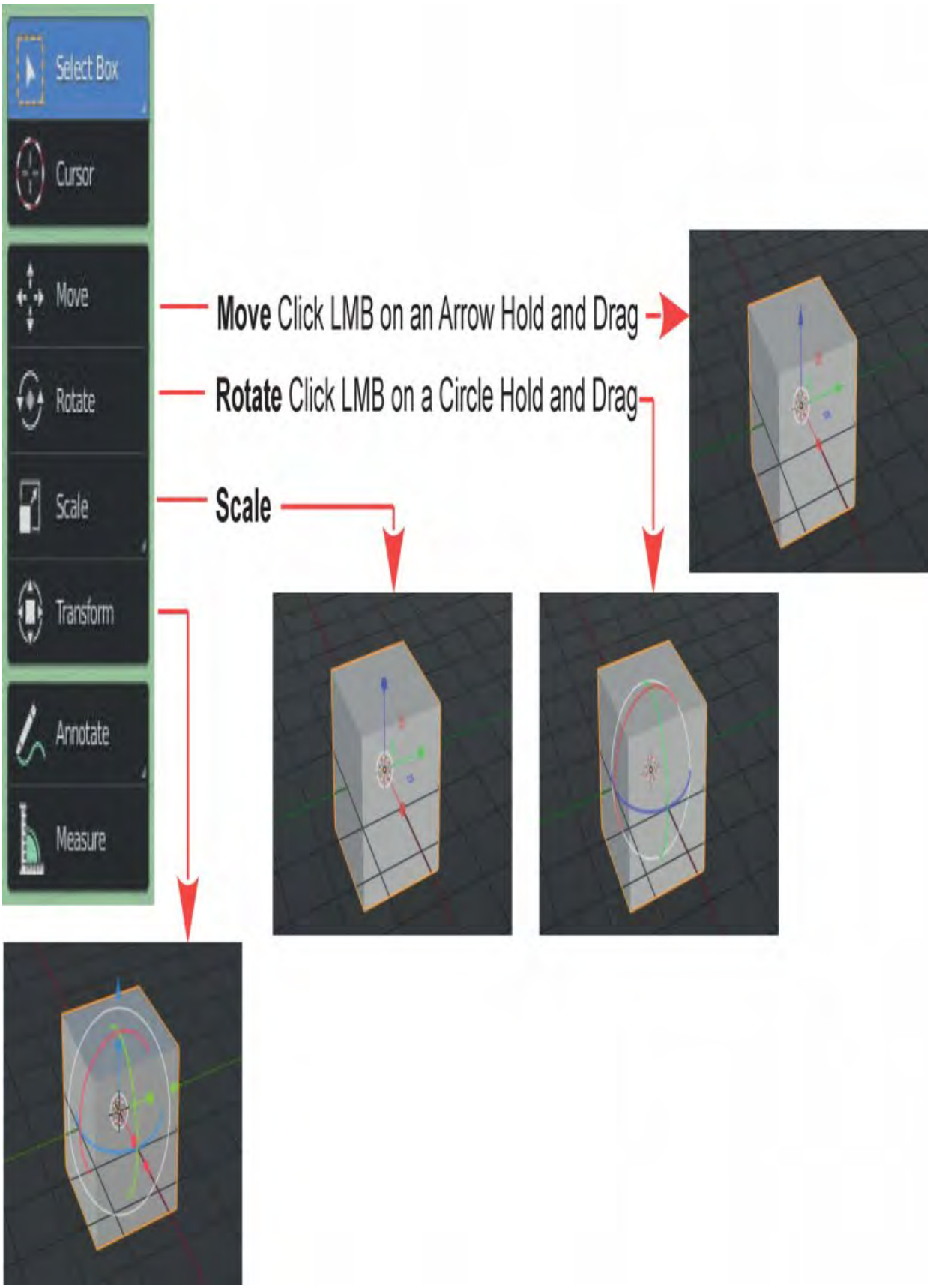

Select Box
Cursor
Move
Rotate
Scale
Transform
Annotate
Measure
Move Click LMB on an Arrow Hold and Drag
Rotate Click LMB on a Circle Hold and Drag
Scale

Figure 4.15

Scale Click LMB on a Paddle, Hold and Drag

Note: Click, Hold and Drag a Square to confine the Scale to an Axis.

Transform combines Move, Rotate and Scale

4.15 Manipulation Units

By default Translation and Scaling units are expressed in **Metric** values and Rotation is in **Degrees**. You can change the values in the **Properties Editor, Scene Properties, Units tab.**

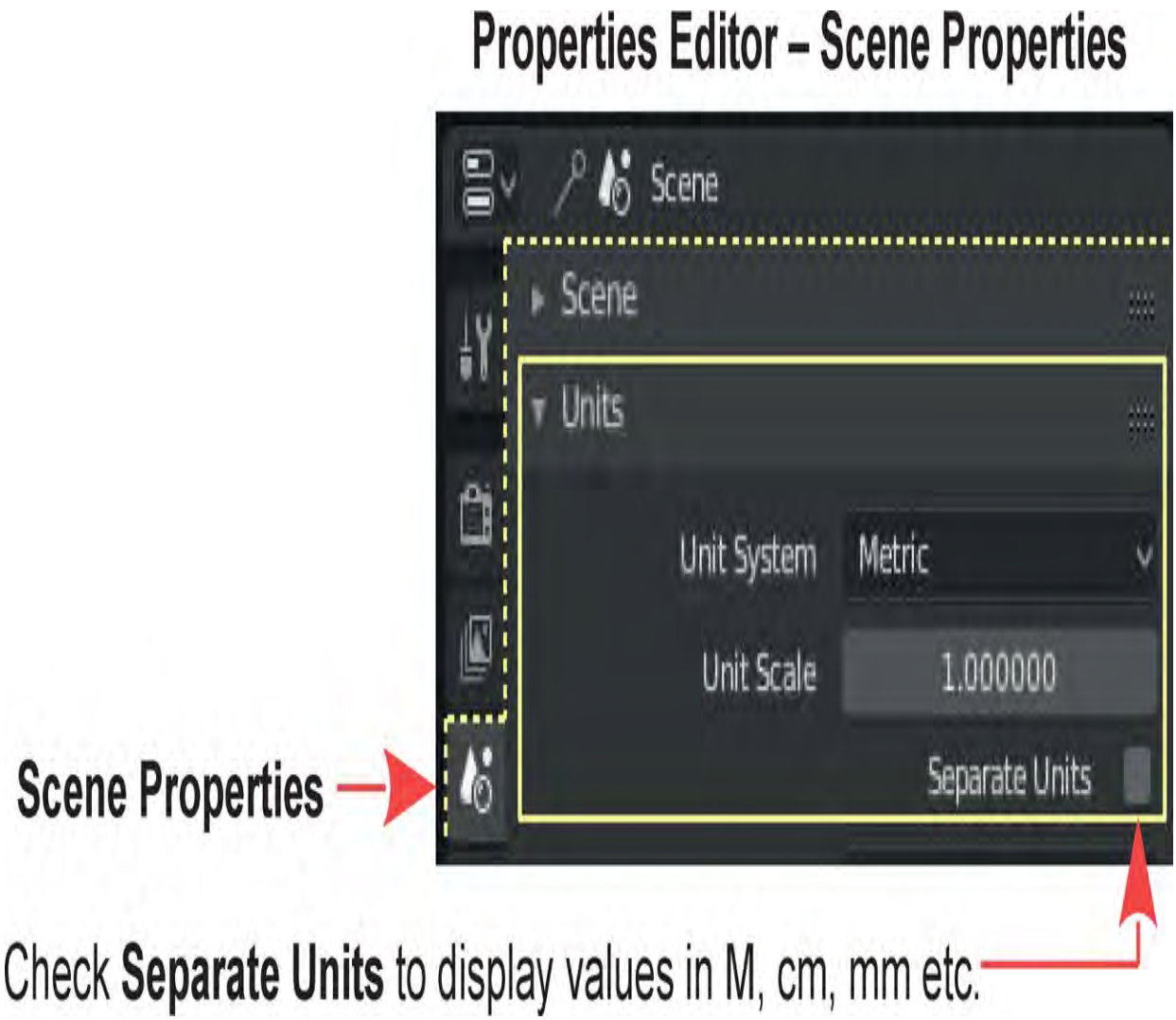


Figure 4.16

When using Metric values the Background Grid in the 3D View Editor is considered to represent One Meter by One Meter divisions.

Scale Units: By default the size of an object is expressed in **Metric Units**. You may elect to change this. In the **Properties Editor, Scene Properties, Units tab** you will see **Unit System** and **Rotation**. Clicking either unit bar will display option menus (Figure 4.17 over).

Unit System – Metric is the default setting, in which case the mid plane grid in the 3D View Editor is considered as representing 1M x 1M units.

Figure 4.17

To demonstrate: With the Mouse cursor in the default 3D Viewport Editor press the G Key (Grab) and move the mouse translating the Cube (the Cube must be selected). In Figure 4.18 the Cube has been translated + 45.89**cm** on the X axis, +1.8844 **M** on the Y axis and +64.109**cm** on the Z axis.

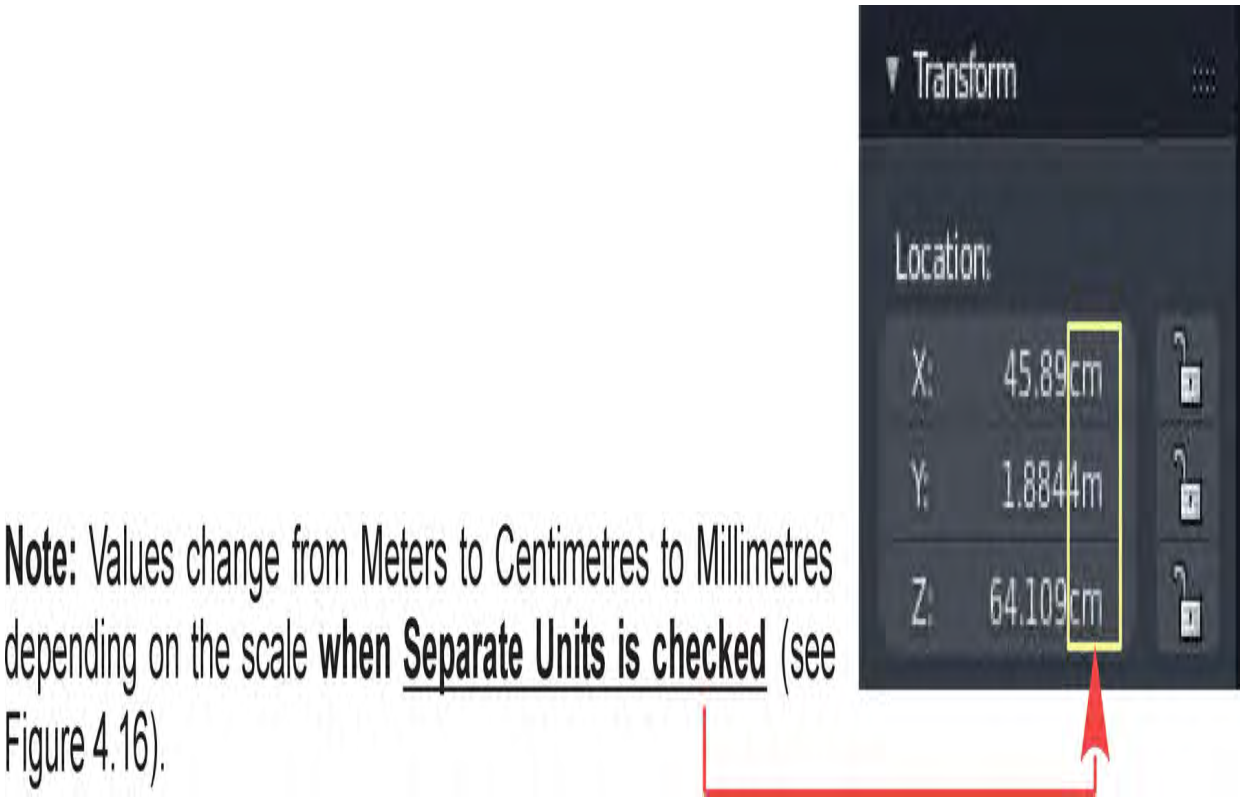


Figure 4.18

You can see the exact values in the **Editor Header** while you have the Cube in Grab mode. You will also see the values in the **Object Properties Panel** (Press the **N Key** to display, Figure 4.18).

None: Changing the **Unit System – Unit Preset** to **None** means you chose to work in Blender units as represented by the division of the mid plane grid (each grid segment = 1 Unit by 1 Unit)

Rotation or Angle: Changing the Unit Preset **Degrees** to **Radians** means you are choosing the measurement where there are **2Tr Radians** in a circle (Figure 4.19n). Angular values are, therefore given in Radians instead of Degrees.

Values in the 3D View Editor display during Manipulation (upper LH corner of the Screen.

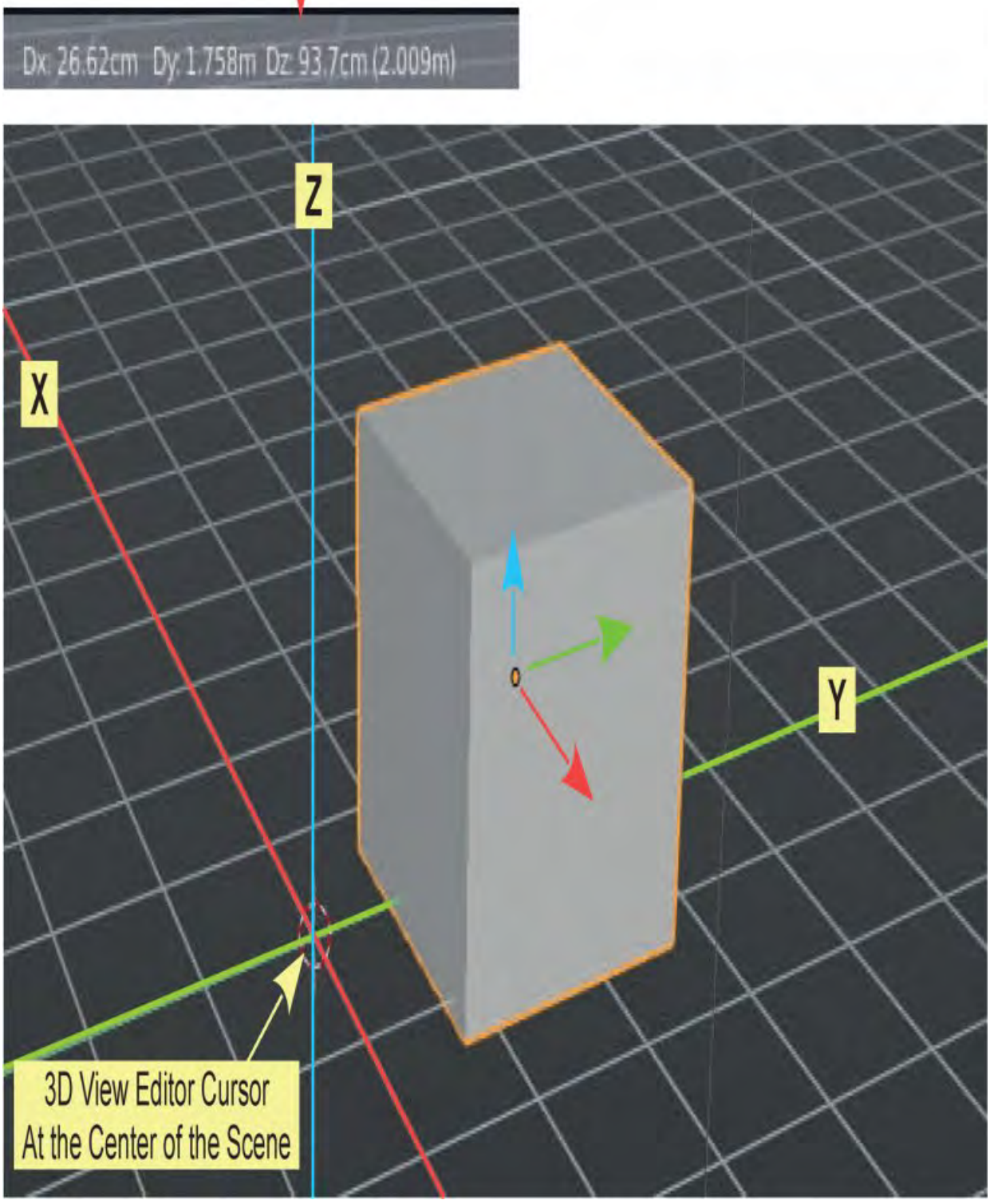


Figure 4.19

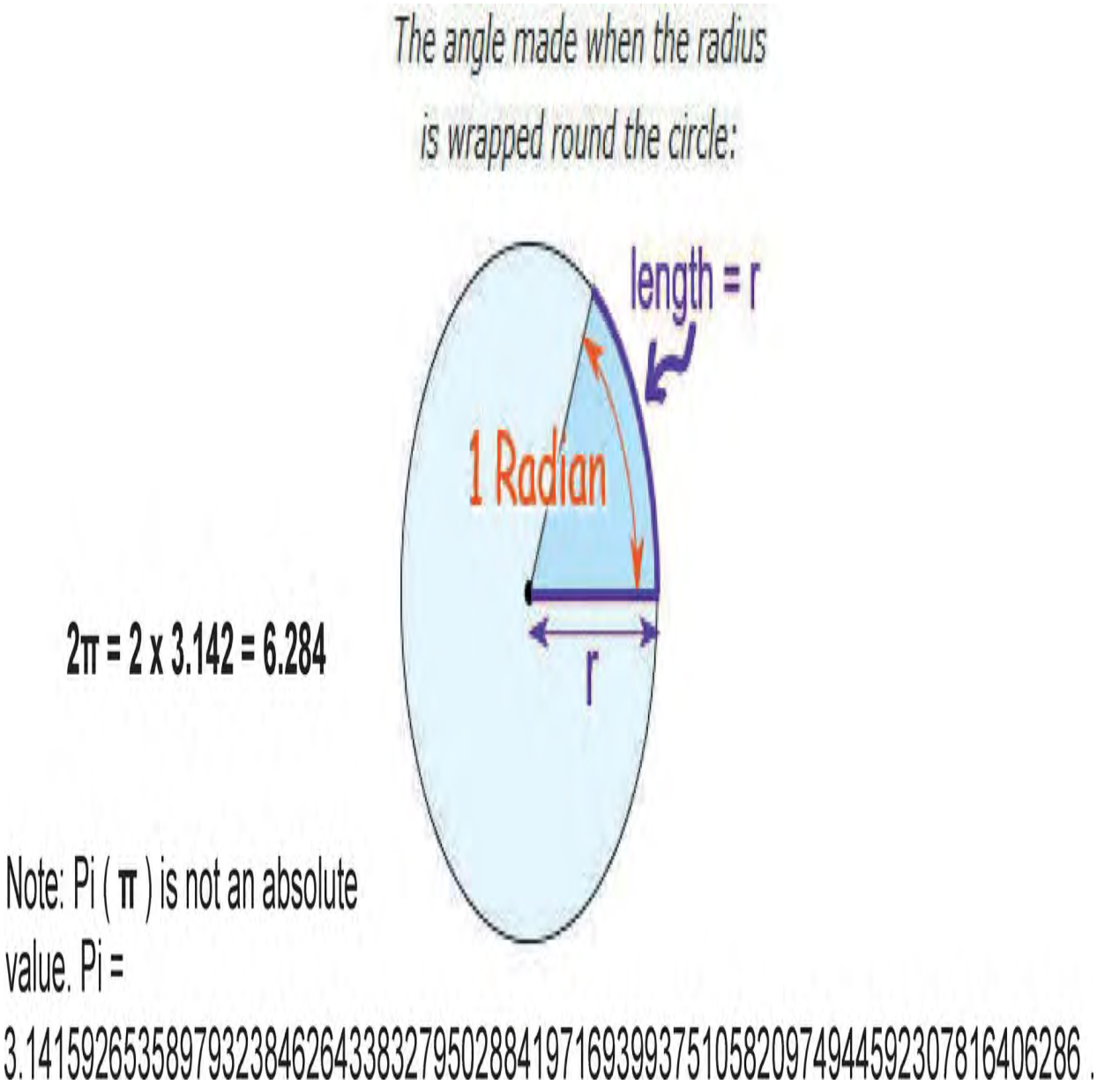


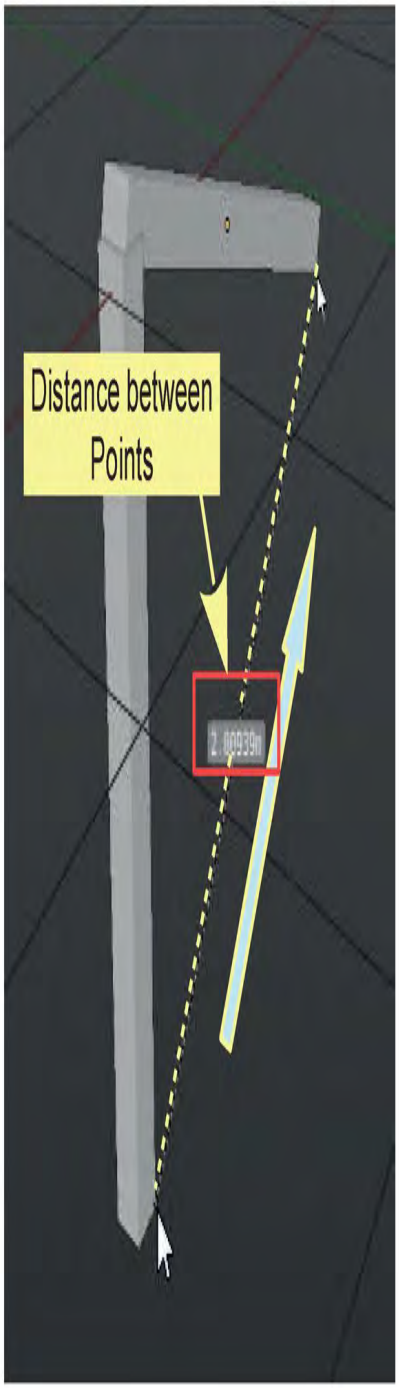
Figure 4.20

4.16 Measuring Ruler / Protractor

At the bottom of the Tool Panel you will see the **Measure button** (Figure 4.21). This button allows you to take liner and angular measurements in the 3D View Editor.

Tool Panel

Linear Measurement



Angular Measurement

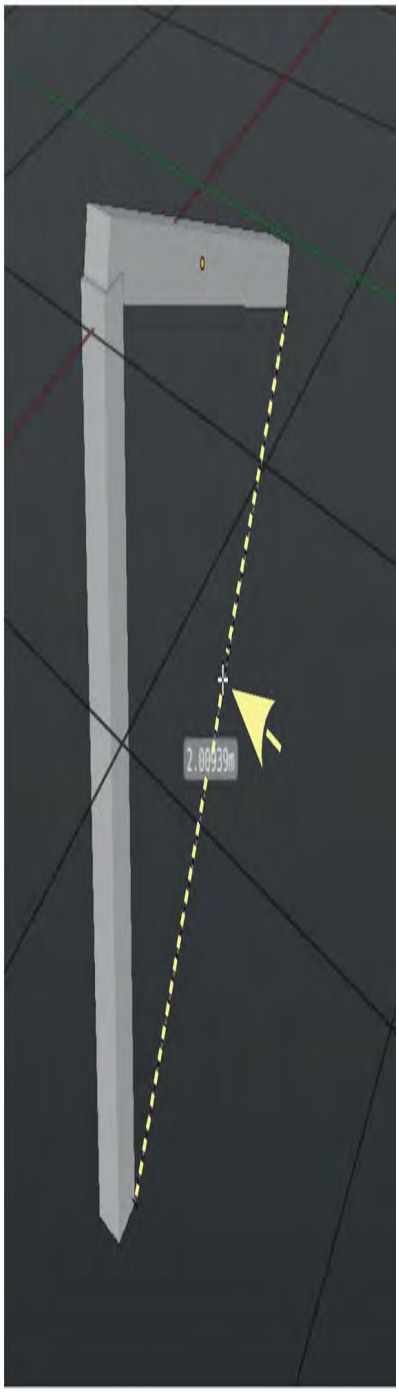
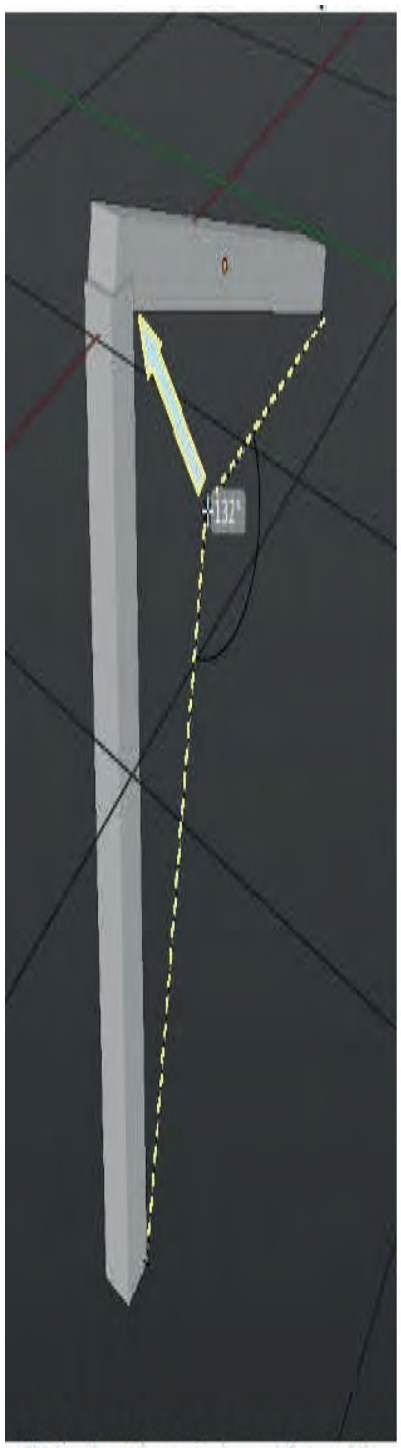
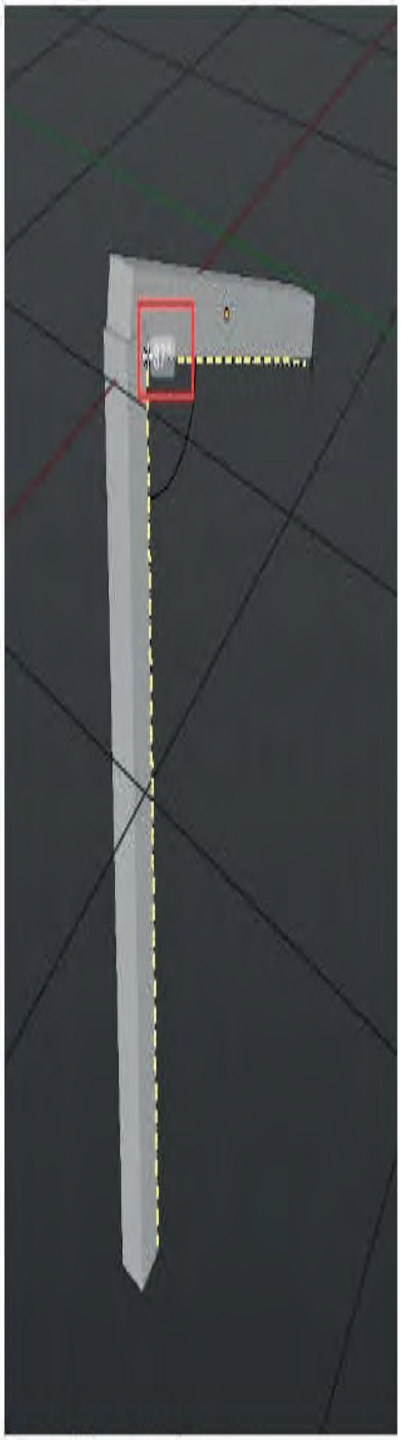


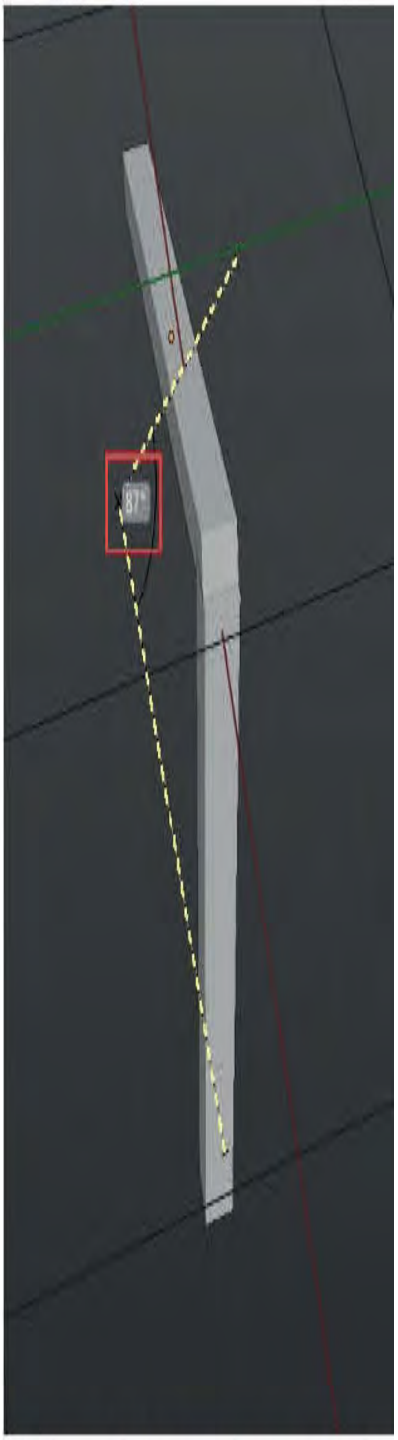
Figure 4.21



Click, hold and drag the mid point to the Apex.



Read the angle.



Plane of the screen rotated showing where measurement were taken (See Note).

Figure 4.22

4.17 Precision Manipulation

Precise Manipulation, precise Translation (location), Rotation and Scaling can be performed in the **3D Viewport Editor, Object Properties Panel, Item Tab** (press the N Key to display). Remember, the values shown in this panel are for the Object that is selected in the 3D Viewport Editor.

In Figure 4.23 the values are for the default Cube Object in its defau position at the center of the Scene. The location is, therefore, X, Y and Z = 0m.

To move the Cube a precise distance away from the center, position the Mouse Cursor over a value. The cursor changes to a double headed arrow.

Click LMB and drag left or right to change the value. The value bar displays the value as you drag the cursor. Release the Mouse button to set the value. The Cube is moved in the 3D View Editor as the value alters. Clicking the **little arrows** incrementally alters the value.

Figure 4.23

You may also click LMB on the value bar to display a typing cursor

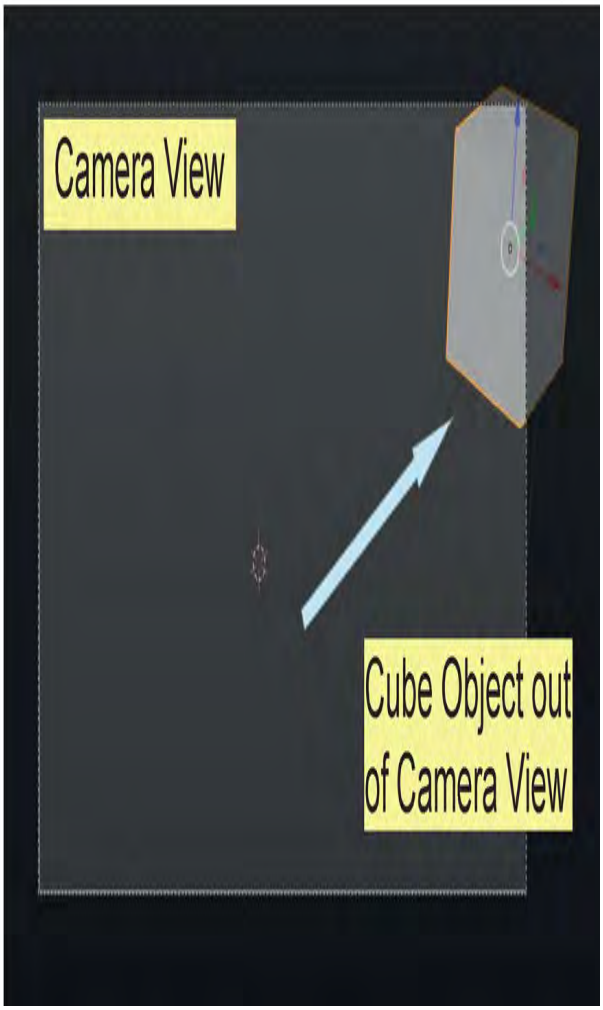


Figure 4.24

Note: When Translating an Object in the 3D Viewport Editor you may be moving the Object out of Camera View. When you render an image your model may not be included. Press **Num Pad 0** to see what is included in the shot.

To return to the original User Perspective View press Num Pad 5 twice, click and hold RMB to rotate the Viewport or press Num Pad 6.

4.18 Coloring Objects

Figure 4.7 shows the **Object Primitives** displayed in different colors. Adding color to an Object in Blender is referred to as; **Applying a Material**. Materials are discussed in Chapter 16 in detail but for the moment consider Material as simply meaning Color.

Color is how you see something in a certain lighting condition. A white ball in the sunlight will look completely different under a colored street lamp at night. This phenomena is true in Blender. How something appears on the computer screen is determined by the simulated lighting effects that are generated, the Material color that is applied and the **Viewport Shading Mode** that is being employed. For the present, how to color an Object in the default Screen arrangement will be demonstrated. The default **Viewport Shading Mode** is; **Solid** with **Studio Lighting** (see Chapter 14)

For the demonstration, add a **UV Sphere** Object (Primitive) to the Scene (Figure 4.25).

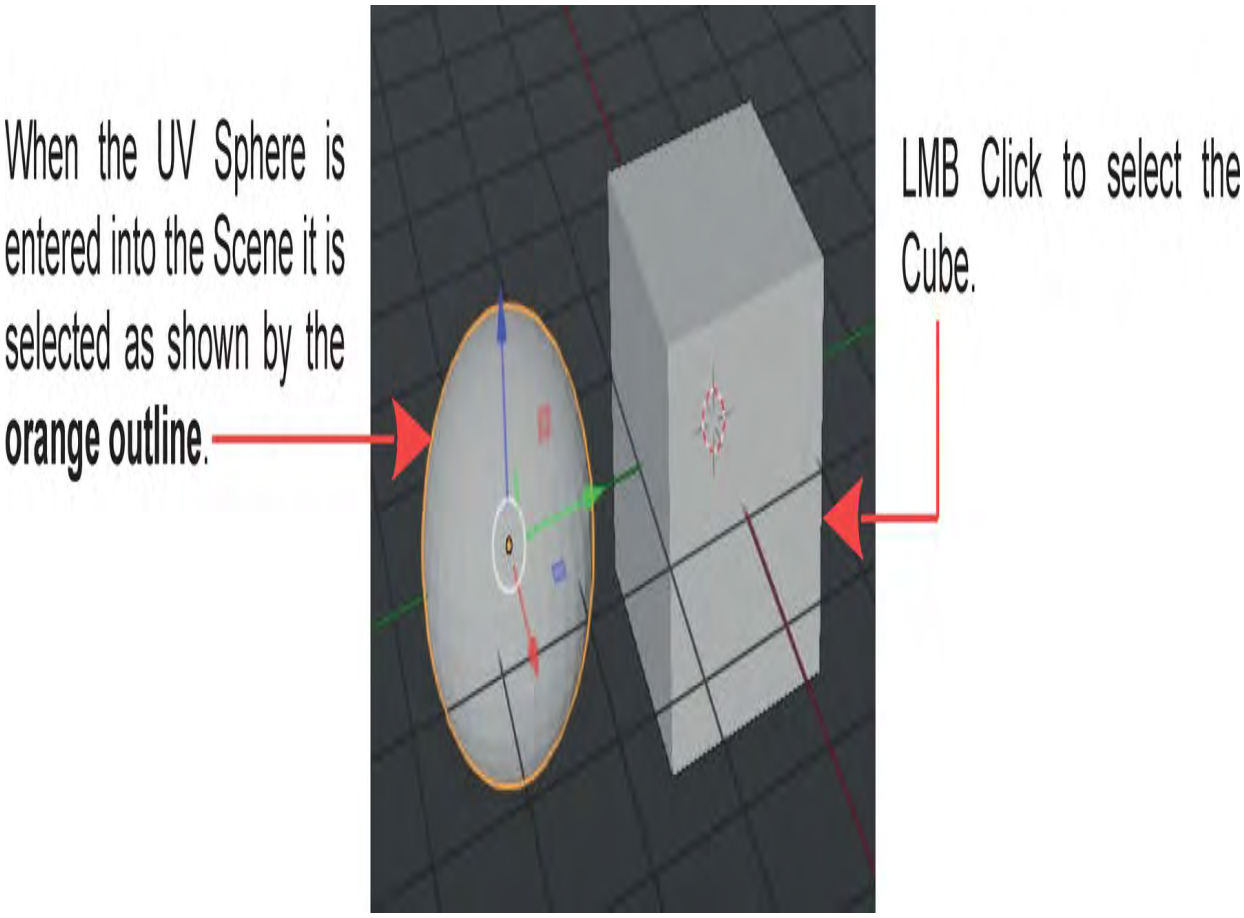


Figure 4.25

Press **Alt + A Key** to deselect or LMB click an empty part of the Editor.

Have the Cube selected in the 3D Viewport Editor then go to the Properties Editor and click on the **Material Properties** (Figure 4.26). In the Properties Editor panel, **Note: Use Nodes** highlighted blue.

Note: With the Cube Object selected the controls in the Properties Editor apply only to the Cube.

Properties Editor
Cube
Material
Material
Node System Active
Highlighted Blue
Click to Disable
Preview
Surface
Use Nodes
Surface
Principled BSDF
Default Gray
GGX
Base Color
Subsurface
0.000
Subsurface Radius
1.000
0.200
Material Properties

Figure 4.26

Properties Editor

The blue highlight indicates that the **Blender Node System** for applying the Material is active. The Base Color shows a color bar with the default gray color of the Cube. Clicking on the color bar displays a color picker circle where you select a different Material (color) **BUT;** If you are reading this book for the first time and are progressing chronologically you have not encountered **Nodes** (Chapter 16) or **Viewport Shading** (Chapter 14) both of which determine whether a new Material will display in the 3D Viewport Editor.

To allow you to add **Material** color at this point, click on the **Use Nodes** bar (the blue bar). This disables the Node System and allows a basic Material to be added to an Object and to display in the default 3D Viewport Editor.

In the **Surface Tab** click on the **Base Color bar** to display the **Color Picker**, click in the circle to select a color. The color is applied to the Cube in the 3D View Editor. Note that the brightness slider at the side of the color picker is towards the top making the applied color comparatively bright.

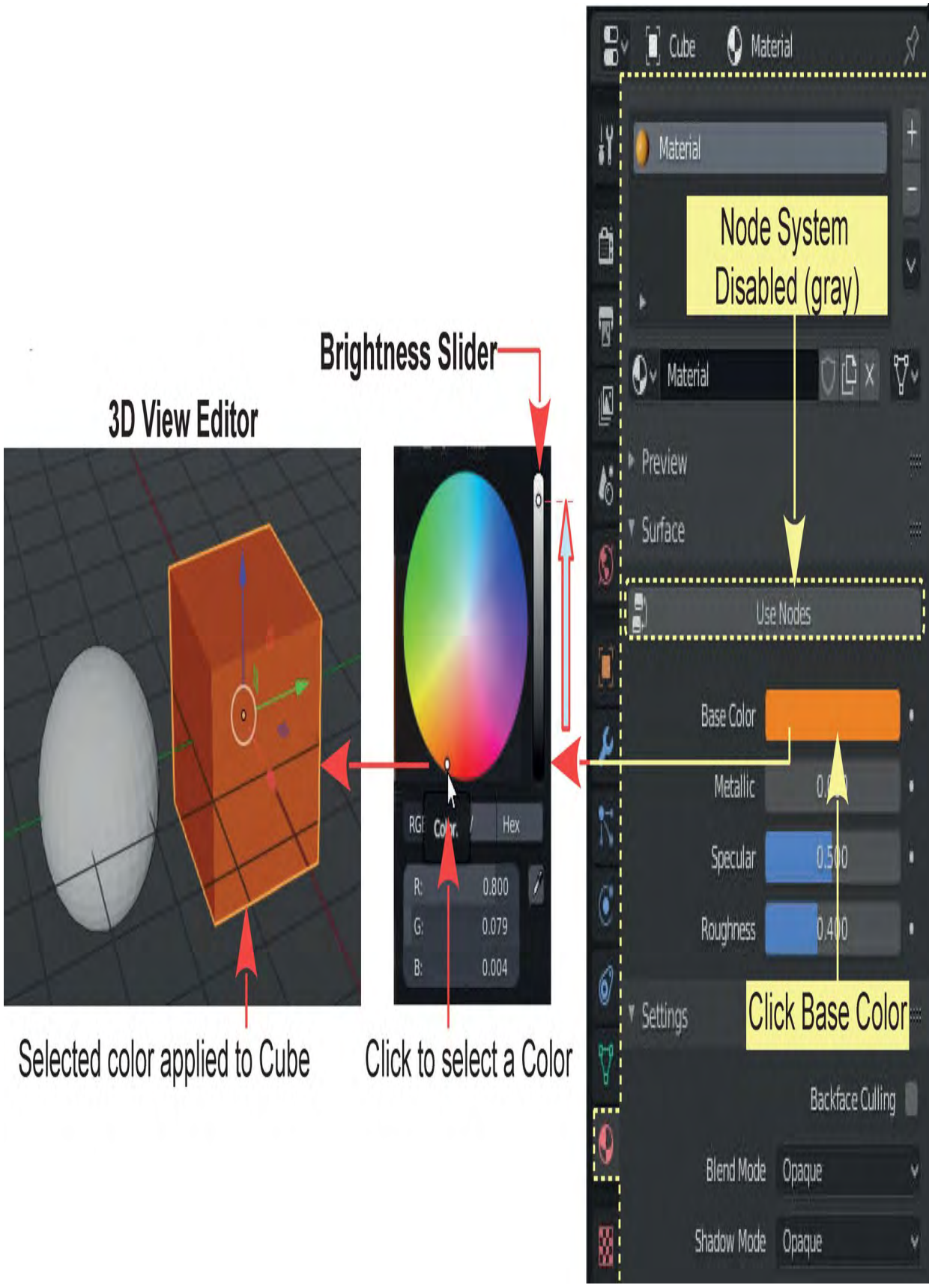

Cube
Material
Material
Node System
Disabled (gray)
Material
Preview
Surface
Use Nodes
Base Color
Metallic
Specular
Roughness
Settings
Click Base Color
Backface Culling
Blend Mode
Opaque
Shadow Mode
Opaque
Brightness Slider
3D View Editor
Hex
R:
0.800
G:
0.079
B:
0.004
Selected color applied to Cube
Click to select a Color

Figure 4.27

Note: At this point the color will only display with the Node System disabled with the 3D View Editor in Solid Viewport Shading Mode. With color applied to the Cube, deselect the Cube and select the UV Sphere.

When you select the **UV Sphere** the **Properties Editor Material Properties** change showing a minimal display containing only the **New** button (Figure 4.28).

Figure 4.28

Click the **New** button to open Material (color) controls for the **UV Sphere**.

In the default Scene, the default Cube Object displayed with a gray Material color. This color is a **default Material** which shows on all new Objects entered into a Scene. In the case of the default Cube the Material is

considered to be pre-applied. In the foregoing exercise, coloring the Cube has modified the pre-applied gray Material.

When a new Object is entered into a Scene it displays with the default gray color but the Material **HAS NOT** been applied. Blender needs something to show, therefore, the default gray is used.

You click the **New** button in the **Properties Editor, Material Properties** to apply the default gray Material. You then modify the Material. With the UV Sphere selected, clicking **New**, again, activates the Node System, reverting to the **Base Color** used previously.

In the new **Properties Editor, Material Properties** click in the bar where you see **Use Nodes** to **cancel the Node** The Editor will display the arrangement similar to that employed for the Cube Object.

Color Gray selected
RGB
HSV
Hex
R 0.800
G 0.800
B 0.800
A 1.000
Note: RGB Color values.RGB: 0.800 produce gray.
A:1.000 = Alpha (Transparency)
Sphere
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Click Base Color
Base Color
Metallic 0.000
Specular 0.500
Roughness 0.400
Settings
Backface Culling
Blend Mode Opaque
Shadow Mode Opaque
Clip Threshold 0.500
Screen Space Refraction
Material Properties–Nodes Cancelled
Sphere
Material.001
Material.001
Material.001
Click to Cancel Nodes
Surface
Use Nodes
Surface Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius 1.000
0.100
Subsurface Color
Metallic 0.000
Specular 0.500
Material Properties
Material Properties–Nodes Active

Figure 4.29

Material Properties–Nodes Cancelled Material Properties–Nodes Active

Clicking **Base Color** and selecting a color applies it to the UV Sphere (the selected Object). **Note:** The Material Properties apply to the selected Object only. With a different Object selected a different Properties panel is displayed.

4.19 Other Types of Objects

Besides **Object Primitives** there are other types of Objects which may be introduced into a Scene. These have a variety of uses and are accessed by clicking the **Add** button in the **3D Viewport Editor Header** or by pressing **Shift + A Key** on the Keyboard (Figure 4.30).

How the different types are used will be explained as you progress through the book but you see the **Mesh** option at the top of the Add menu which opens the list of Primitives previously discussed.

From the menu you will see that you may add additional Cameras and Lights into a Scene. These influence how you see Objects. For one type of special Object see 4.22 **Metashapes** (Metaball).

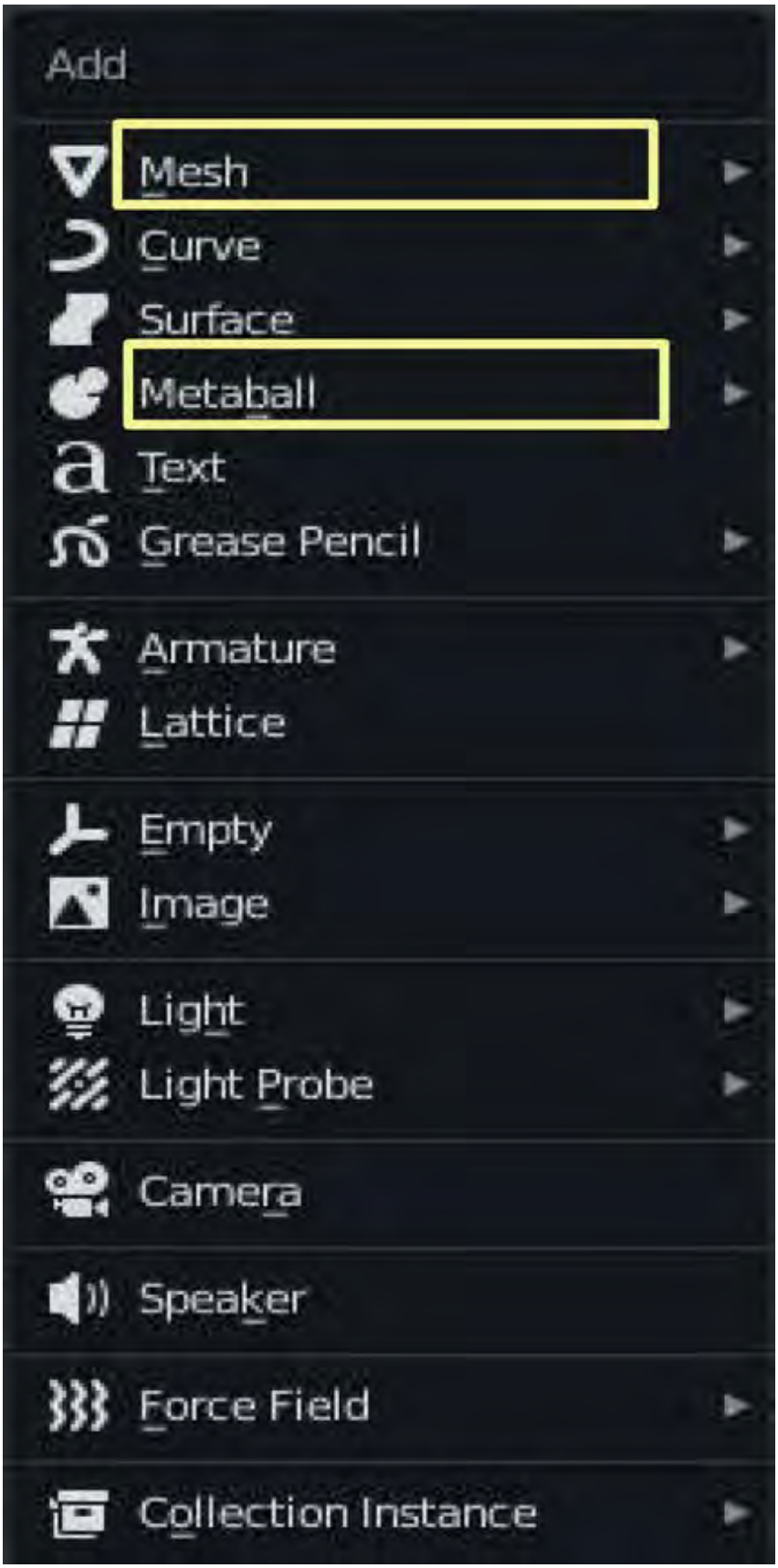

Add
Mesh
Curve
Surface
Metaball
Text
Grease Pencil
Armature
Lattice
Empty
Image
Light
Light Probe
Camera
Speaker
Force Field
Collection Instance

Figure 4.30

4.20 Naming Objects

As you enter Objects to a Scene Blender automatically assigns a name. In Figure 4.31 the Cube Object has been named **Cube**, the Monkey Object has been named **Suzanne** and the UV Spheres have been named **Sphere, Sphere.001** and **Sphere.002**.

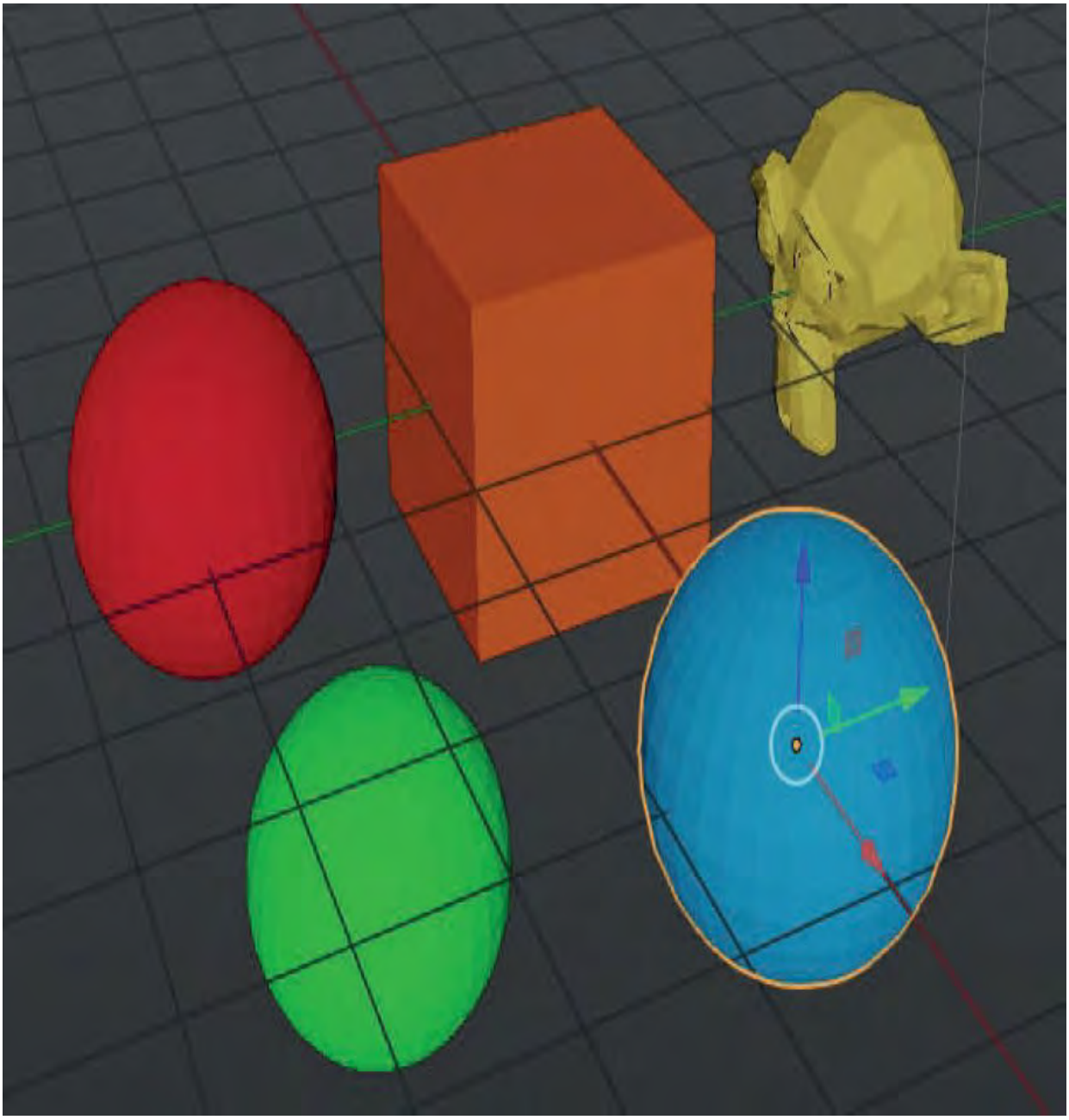


Figure 4.31

The names are displayed in the **Outliner Editor** under **Scene Collection** (Figure 4.32).

Figure 4.32

You will also see an Object's name in the **Properties Editor** with the **Object Properties** selected (Figure 4.33).

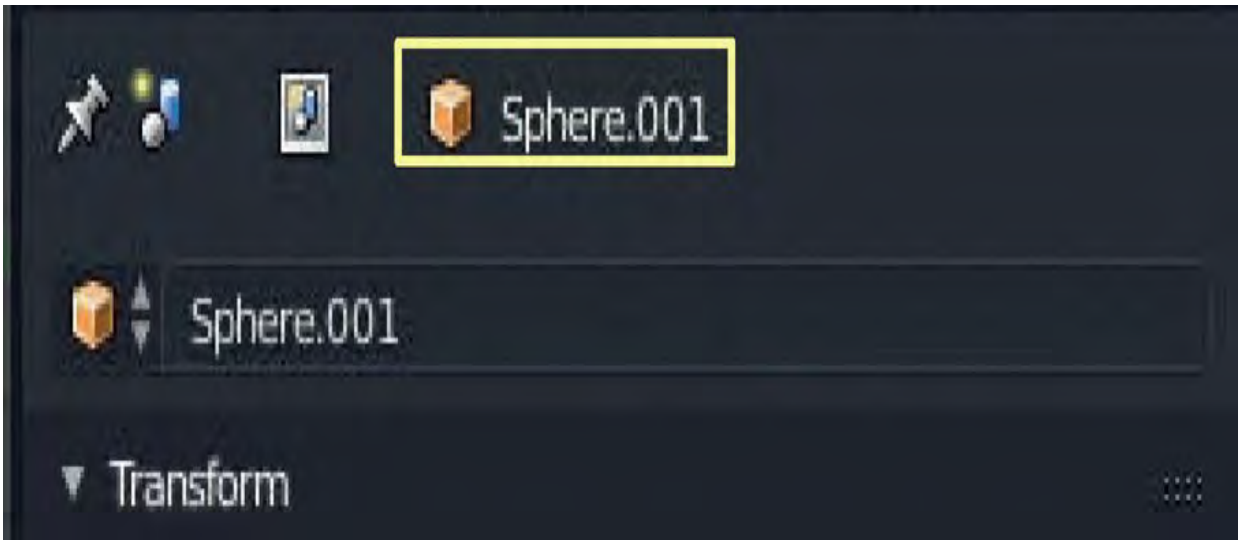


Figure 4.33

The automatic names are all very well, but as you build a Scene and add many Objects, it is obvious that automatic names could become meaningless and confusing. For example; **UV Spheres** are named **Sphere, Sphere.001** and **Sphere.002**.

> Blender has a monkey head that's affectionately referred to as **Suzanne**, a reference to the ape in two of Kevin Smith's films: *Jay and Silent Bob Strike Back* and *Mallrats* (close to the end). Many 3D modeling and animation suites have a generic semi-complex primitive that is used for test renders, benchmarks, and examples that necessitate something a little more complex than a cube or sphere.

The name **Suzanne** will definitely distinguish Monkey unless you add more Monkeys, in which case they will be named Suzanne, Susanne.001, Suzanne.002 etc. Even though the UV Spheres have been colored the question arises, *which one is which?*

Obviously it is preferable to name Objects with meaningful names especially when they have been reshaped into Characters or components of a

complex Scene.

You rename Objects in the **Properties Editor, Object Properties** or in the **Outliner Editor** by double clicking on the Object name, backspacing or deleting the name and retyping a new name.

A handy way of identifying Objects in the 3D Viewport Editor is to check (tick) **Name** in the **Properties Editor, Object Properties, Viewport Display Tab**. The name of the Object is displayed in the Editor (Figure 4.34).

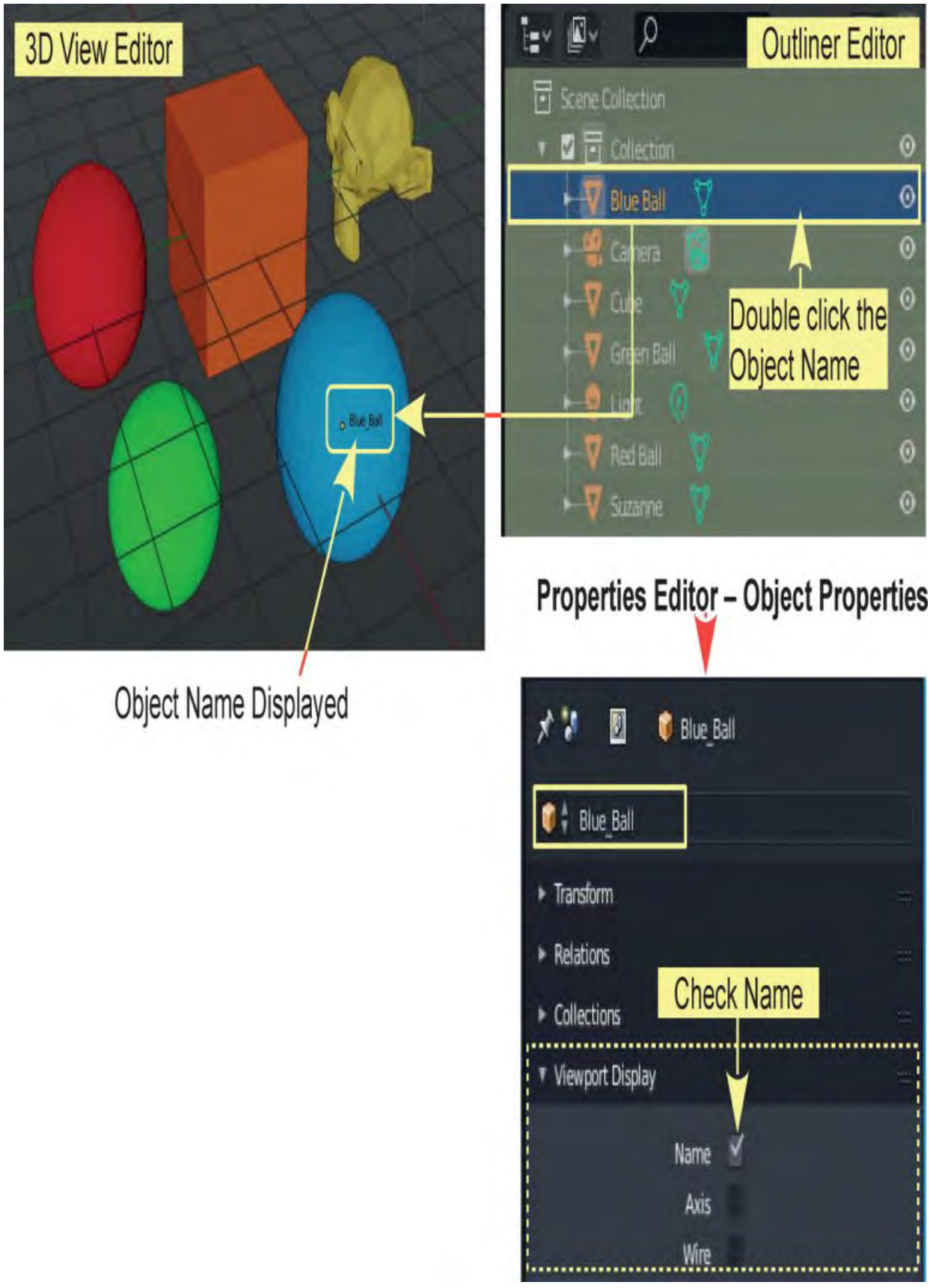

3D View Editor
Outliner Editor
Scene Collection
Collection
Blue Ball
Camera
Cube
Green Ball
Light
Red Ball
Suzanne
Double click the Object Name
Blue_Ball
Object Name Displayed
Properties Editor – Object Properties
Blue_Ball
Blue_Ball
Transform
Relations
Collections
Check Name
Viewport Display
Name
Axis
Wire

Figure 4.34

Note: When Objects are renamed they are alphabetically listed in the Outliner Editor.

Note: Objects in the 3D Viewport Editor are listed in the Outliner Editor in groups called **Collections**. The arrangement and management of Collections are detailed in Chapter 12.

4.21 The Header Button Menus

The 3D Viewport Editor **Header,** buttons provide menus for selecting a variety of functions. The individual functions will be called upon as required in specific instructions. Some functions have already been covered such as switching between Object Mode and Edit Mode, Object Selection (Select button) and adding Objects (Add button). The menus allow you to activate a function by clicking the function name and in many cases provide a Keyboard shortcut. For example; in the Select button menu you will see Box Select (Tweak Left – LMB Clk Drag) and Select Circle (C Key). You either click the select type name in the menu or press the Keyboard shortcut with the Mouse cursor in the 3D Viewport Editor panel. The Add button displays the Object Add Menu which is the same as pressing Shift + A Key. The Object button displays a comprehensive menu with sub menus. As you progress in Blender you will become familiar with the functions in the menus but for now, the following are worth mentioning:

Join (Ctrl + J Key)

Selecting two or more Objects at the same time (Hold Shift + LMB Click) then clicking Join in the menu or pressing Ctrl + J Key joins the selected Objects into a single Object.

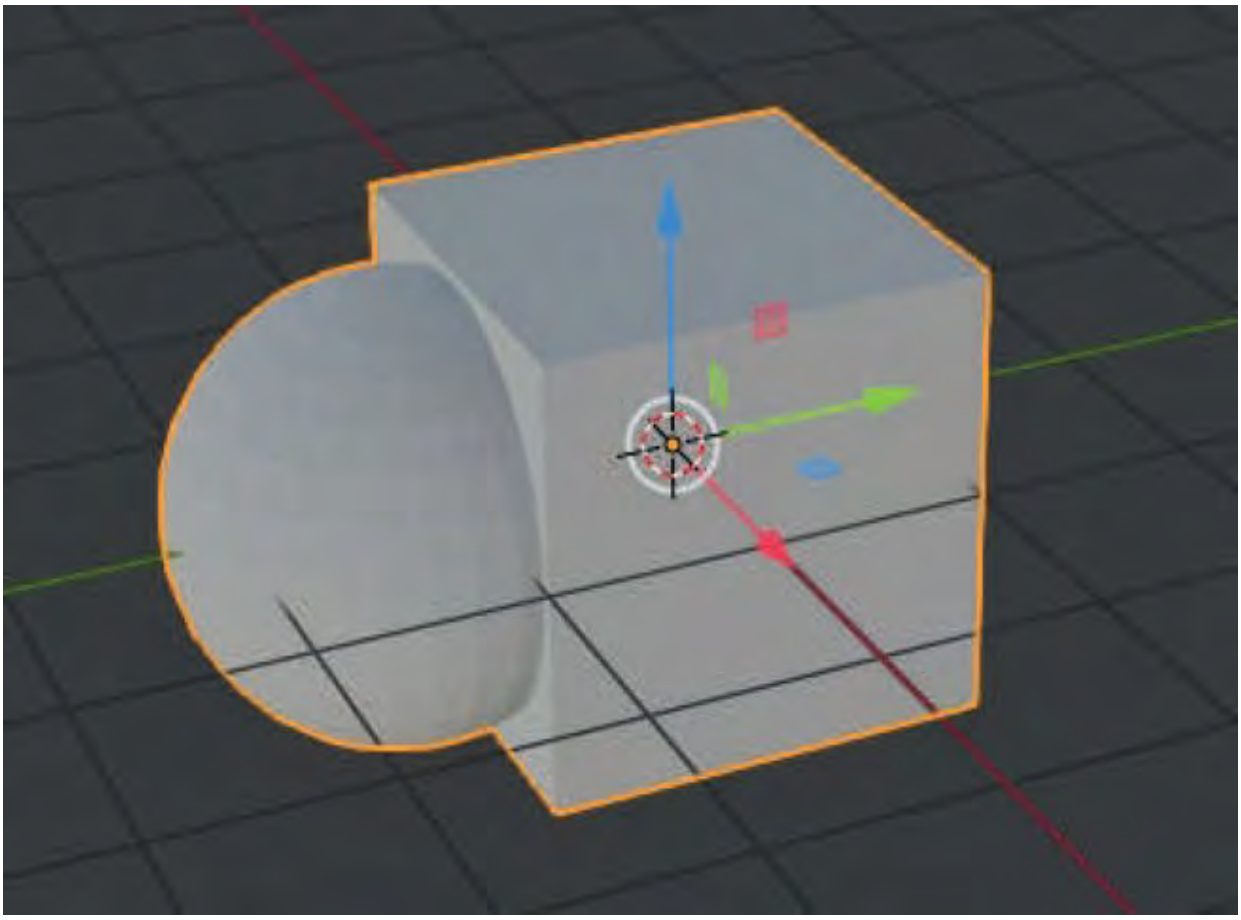


Figure 4.35

Copy and Paste - Objects may be copied from one Blender file and pasted into another file.

Smooth and Flat Shading

In the previous diagrams describing naming and coloring Objects you will observe that the surface of the UV Sphere is made up from a series of rectangular flat surfaces (Flat Shading). In the **3D Viewport Editor Header** selecting **Object** – **Shade Smooth** in the menu produces a smooth spherical surface which is much nicer for coloring the sphere. Alternatively RMB click in the 3D Viewport Editor to display the **Object Context Menu** with the shading options.

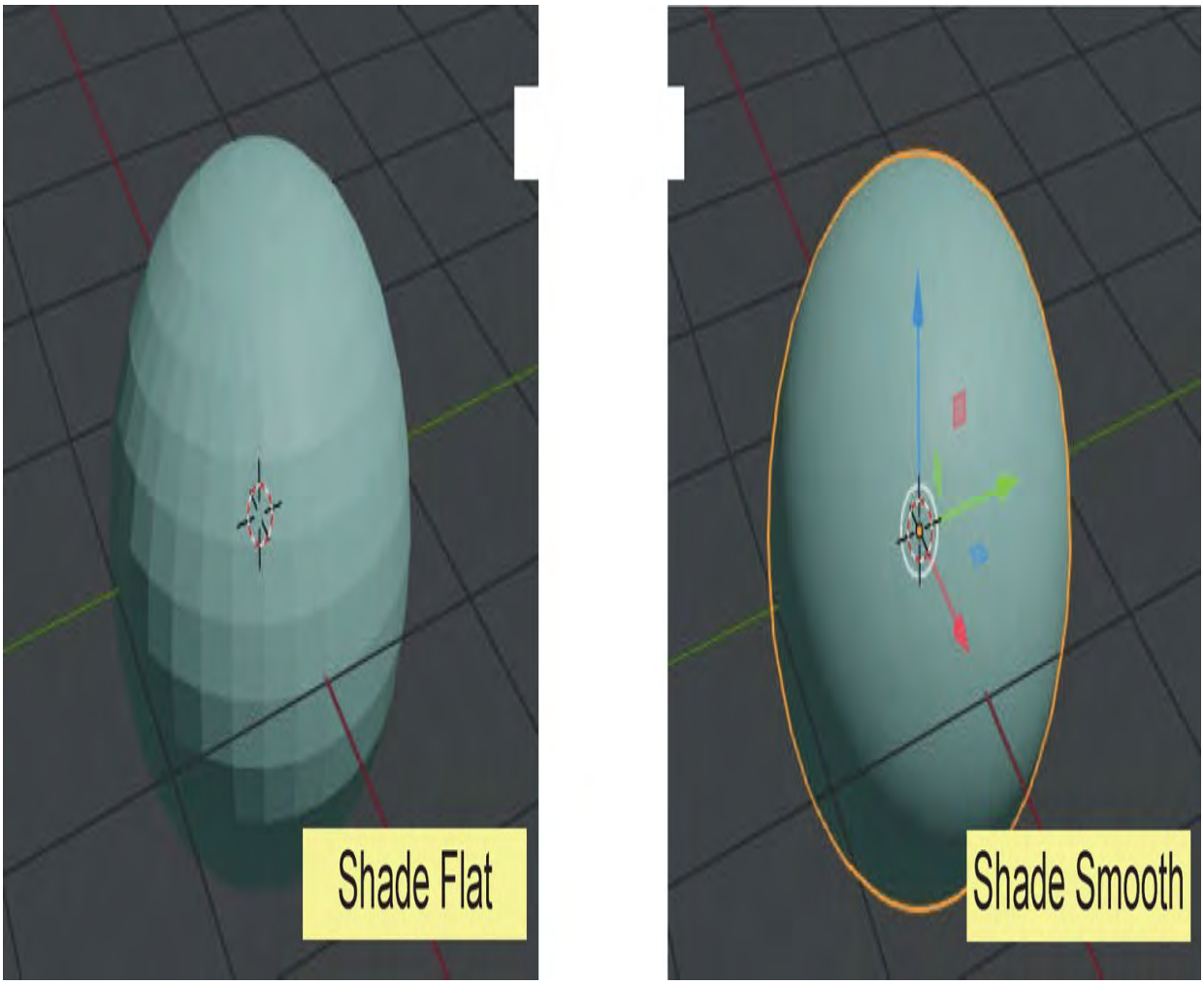


Figure 4.36

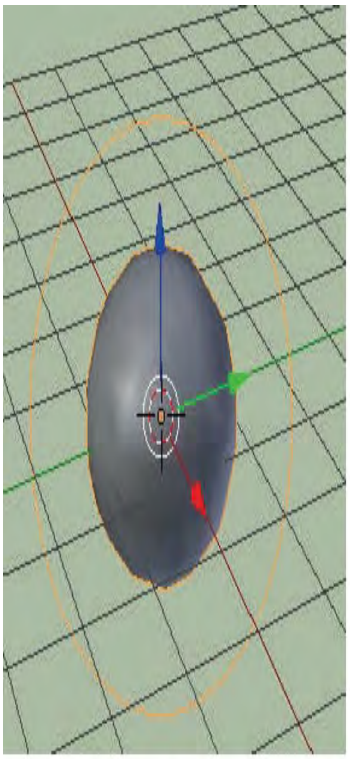
4.22 Meta Shapes

Meta Shapes: Are described as *mercurial, or clay-like forms that have a rounded shape.*

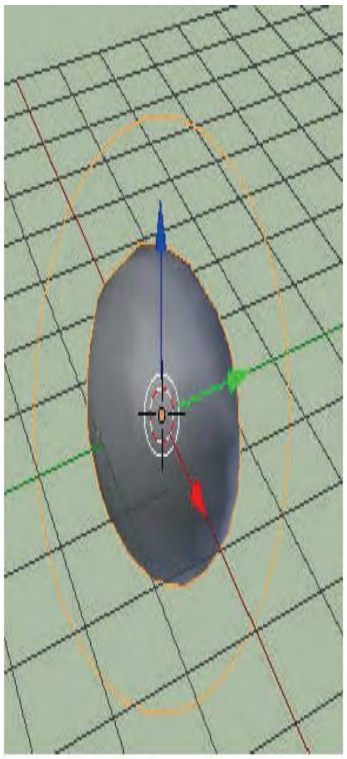
When two **Meta Objects** (Shapes) get close, they begin to interact with one another. They merge, as water droplets do. When they are moved apart, they restore their original shape.

There are several **Meta Shapes** you can use in Blender (Figure 4.37). Meta Shapes are added to a Scene in Object mode like any other shape: press **Shift + the A key – Add – Metaball** and select either Ball, Capsule, Plane, Ellipsoid, or Cube. Be sure to deselect one shape before adding another, or

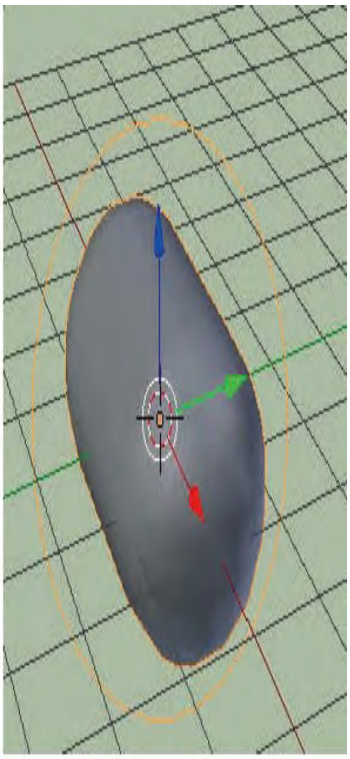
they will be automatically joined. When **Meta Shapes** get close to one another, they begin to pull and flow together like droplets of liquid (Figure 4.38). The shapes can be animated and textured, and reflection and transparency can be applied to create some stunning effects.



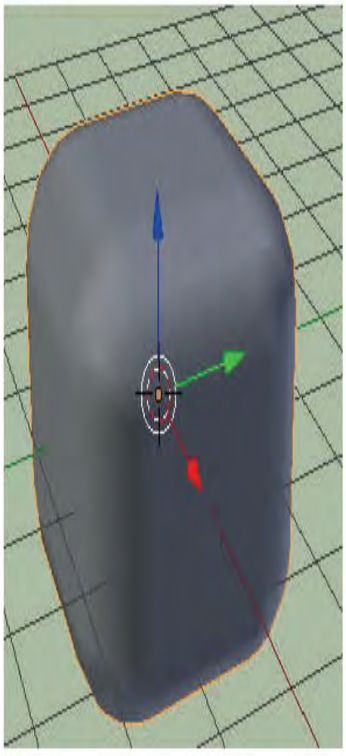
Meta Ball



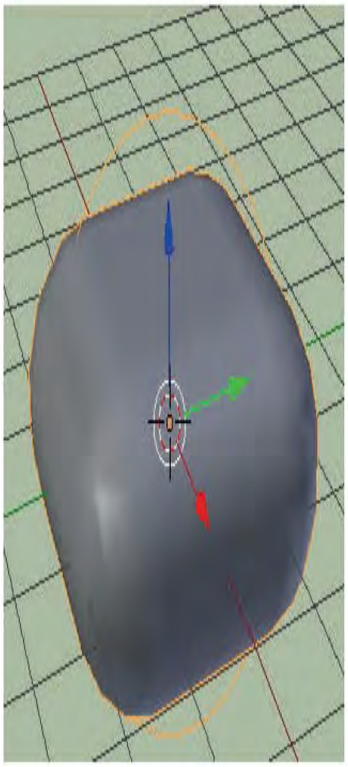
Meta Elipsoid



Meta Capsule



Meta Cube



Meta Plane

Figure 4.37

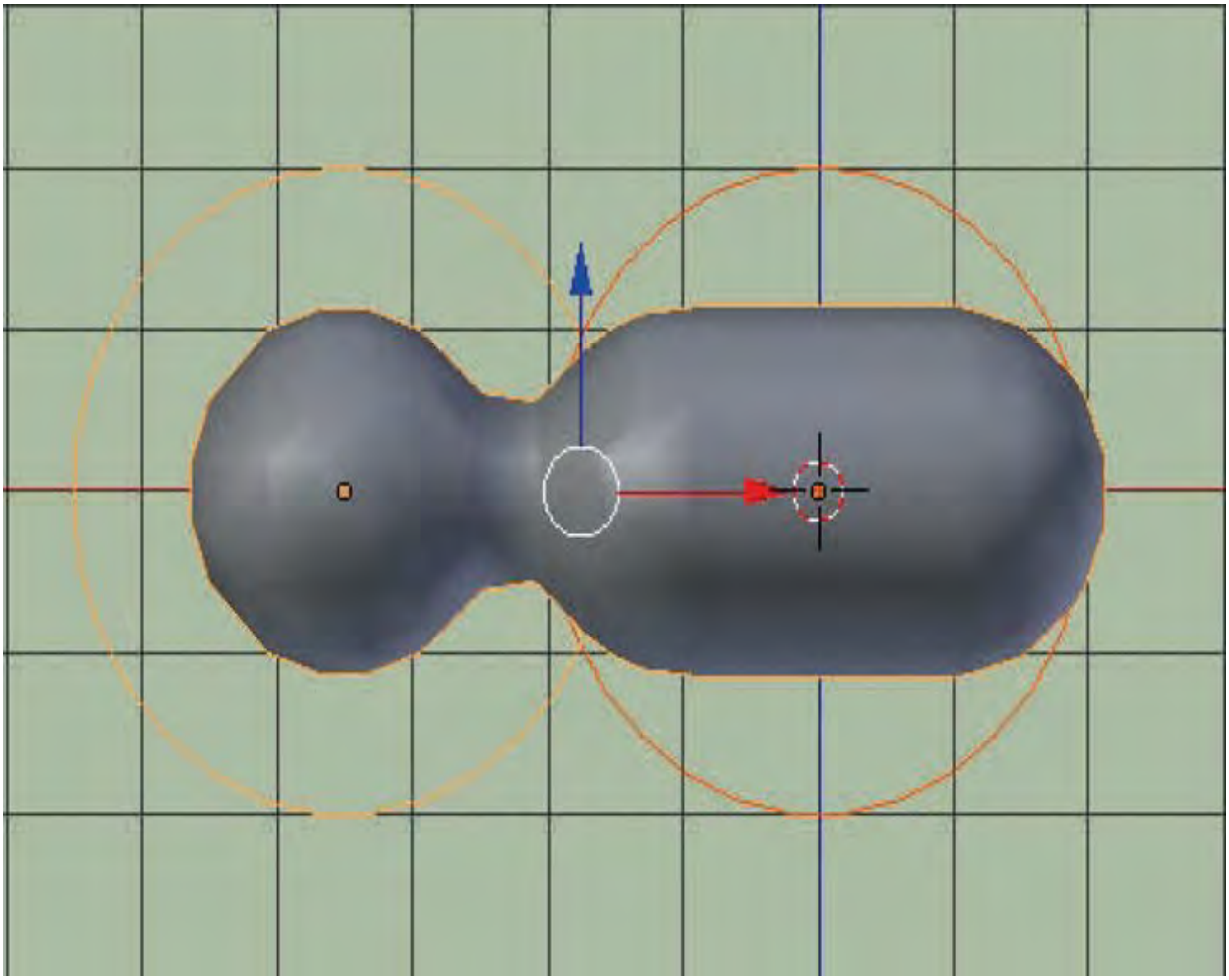


Figure 4.38

Meta Ball and Meta Capsule flowing together when in close proximity.

5 Editing Objects

5.1 The Mesh Object
5.2 Edit Mode Selection
5.3 Selecting Verts Edge Face
5.4 Manipulating
5.5 Creating Vertices
5.6 Adding and Deleting
5.7 Center Points
5.8 Joining and Separating
5.9 Creating Vertex Groups
5.10 Proportional Vertex Edit
5.11 Inset Faces
5.12 Parenting

Editing Objects

Editing or modifying one of the basic shapes (Primitives) in Blender is the process of Modeling.

Creating a model begins by introducing a Primitive to the Scene. Blender's Primitives are Mesh Objects, that is to say, they are constructed with surfaces formed by a mesh. The mesh can be imagined as a fishing net or a piece of chicken coop wire with strands of twine or wire crisscrossing and joined where they intersect. The spaces between the strands are filled in forming a surface.

The shape of a primitive is altered by manipulating the mesh. This is achieved by selecting (grabbing) the intersection points (**Vertices**) or the

strands (**Edges**) or the filled in pieces (**Faces**) and moving them in 3D Space.

Vertices, Edges and Faces can be extruded to build onto a Primitive. Edges and Faces may be scaled and rotated to twist the shape of the Primitive.

Several Primitives can be joined together to shape a single Object.

Suzanne is a Mesh Object (Model)

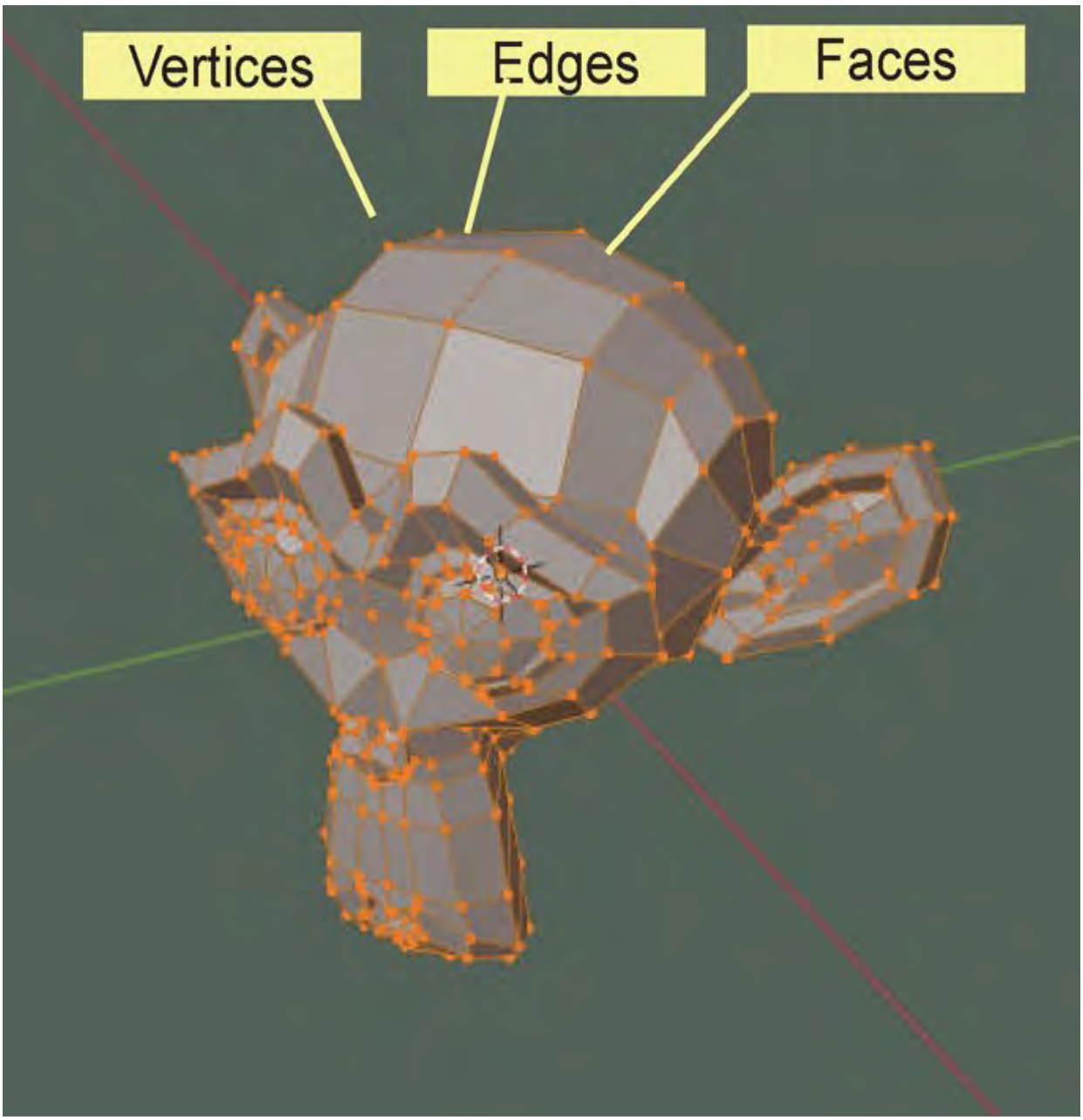


Figure 5.1

5.1 The Mesh Object

The default Blender Scene contains a **Cube Mesh Object** which, by default, is selected in **Object Mode**. With the Cube selected (orange outline) press the **Tab key** to enter **Edit Mode** to see the basic components of the Cube Mesh (Figure 5.2).

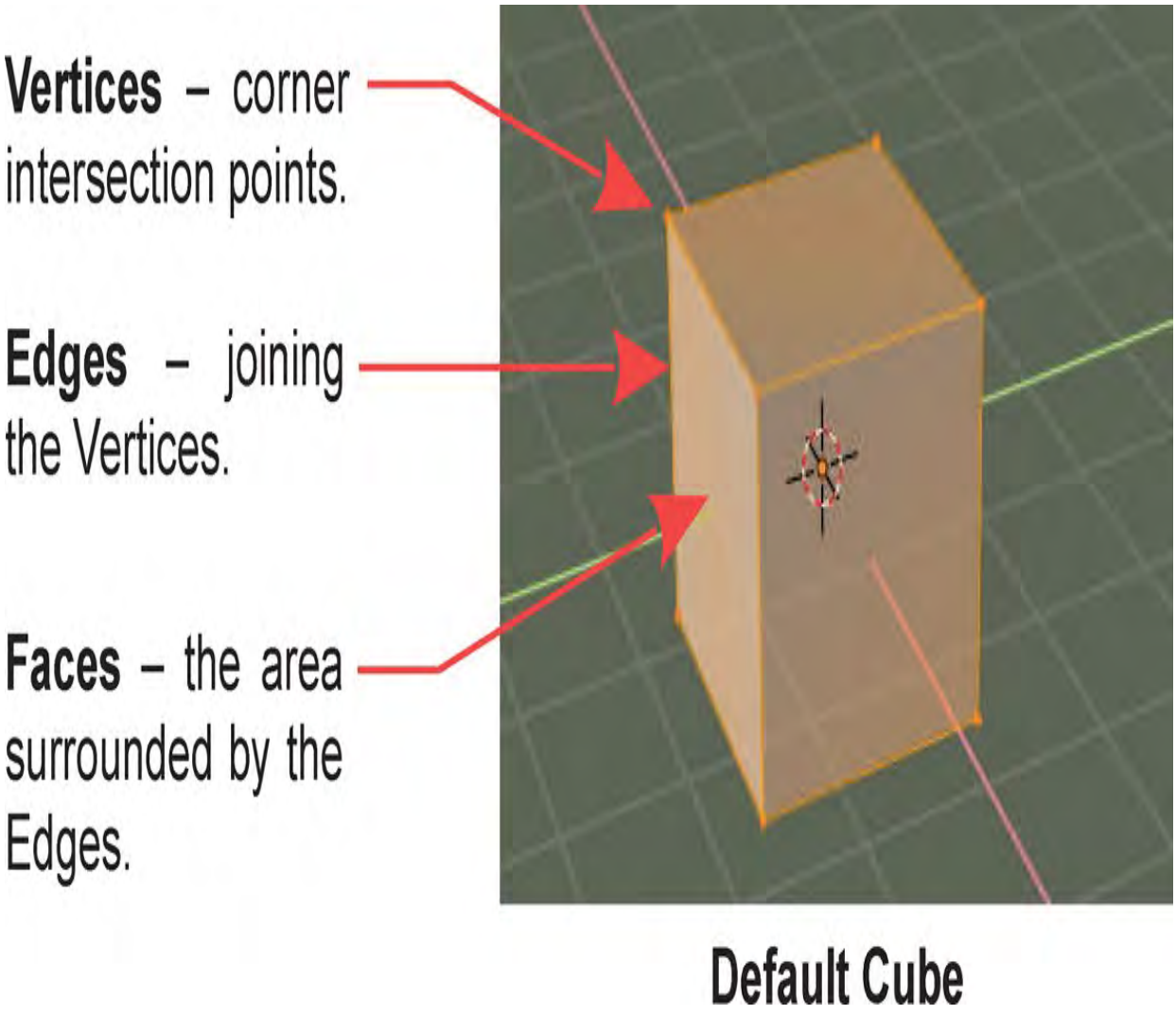


Figure 5.2

5.2 Edit Mode Selecting

With the 3D Viewport Editor In **Edit Mode**, you work with the individual **Vertices** (mesh intersections) to Model the shape. You know you're in Edit

Mode when you see orange lines and dots on the selected Object (Figure 5.2). When you tab into Edit Mode, the whole of your selected Object is in Edit Mode with all the Vertices selected. By default, Edit mode is in **Vertex Select Mode**.

Selection Options

In **Edit Mode**, the default selection Mode is **Vertex** which means you may select Vertices. You can elect to enter **Edge** or **Face** select Mode (Figure 5.3). These options are available in the 3D Viewport Editor Header.

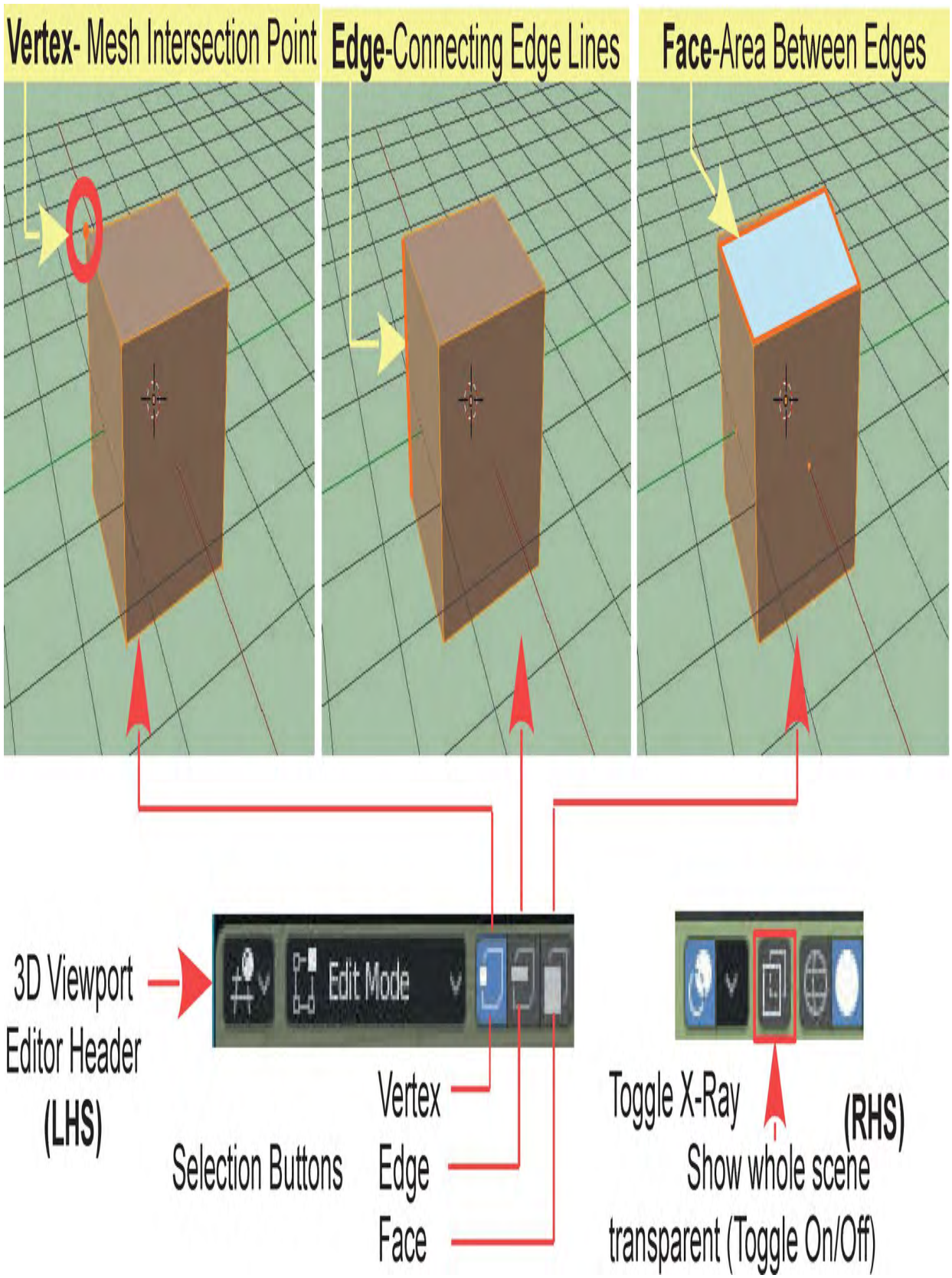
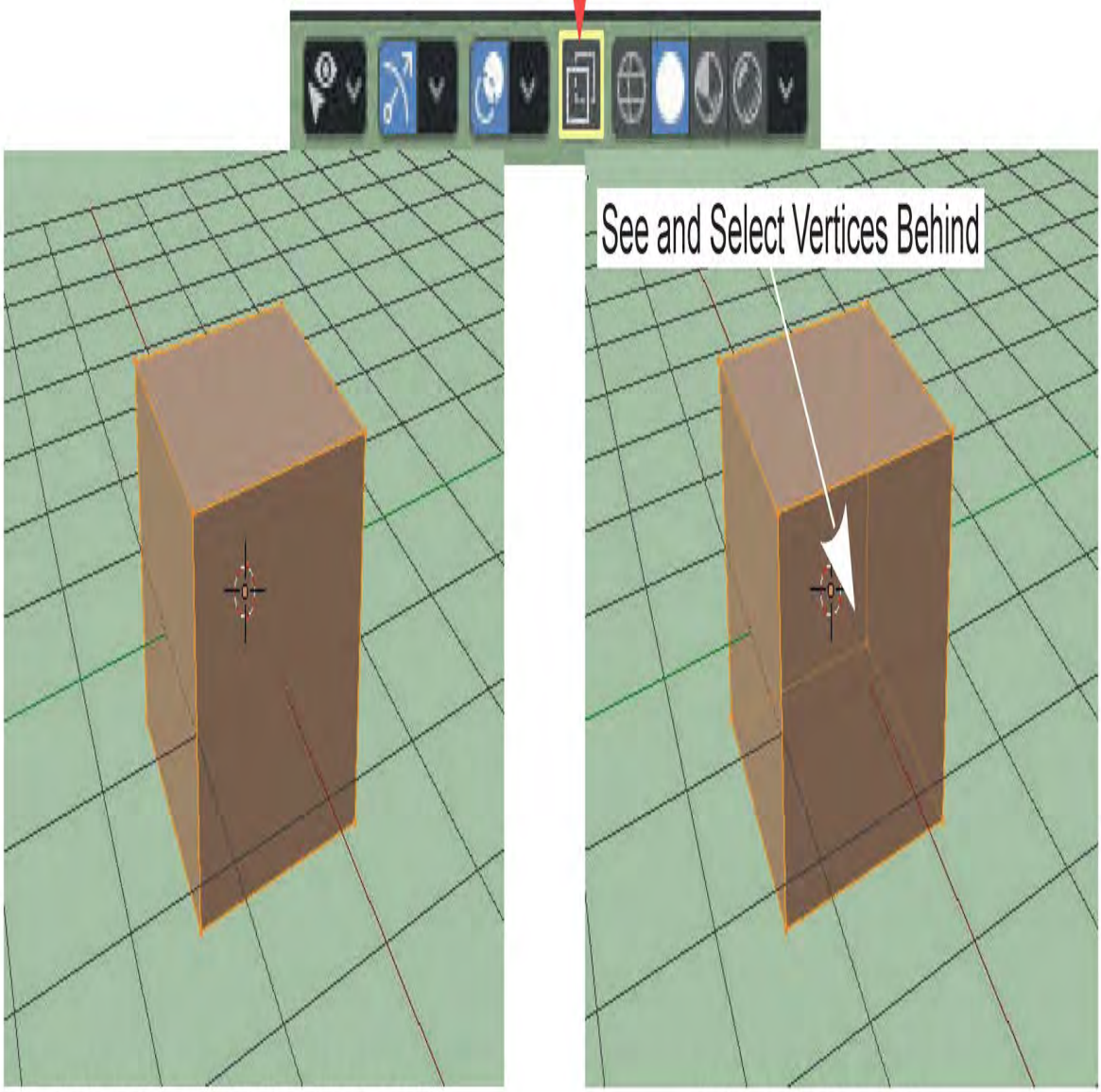

Vertex- Mesh Intersection Point
Edge-Connecting Edge Lines
Face-Area Between Edges
3D Viewport
Editor Header
(LHS)
Edit Mode
Selection Buttons
Vertex
Edge
Face
Toggle X-Ray
(RHS)
Show whole scene
transparent (Toggle On/Off)

Figure 5.3

Also, by default, only visible Vertices, Edges or Faces are available for selection. This means that you can only select the Vertices, Edges or Faces on the front that you actually see in the Editor. Blender has a **Toggle X -Ray** function, which allows you to see and select hidden Vertices, Edges or Faces (Figure 5.4). This function is toggled on and off in the 3D Viewport Editor Header by clicking the **X-Ray** button.



Limit Selection to Visible **X-Ray Off**

Select all **X-Ray On**

Figure 5.4

Toggling **X-Ray On** allows you to see Vertices in Edit Mode which are hidden behind front Faces. You should be aware that this feature exists for Object Mode. You can not select Vertices in Object Mode but there are occasions when you may wish to see hidden geometry.

The 3D Viewport Editor displays Objects in several different Display methods referred to as **Viewport Shading Modes** (see Chapter 14 – Viewport Shading). The default display is Solid Viewport Shading. The options are accessed in the 3D Viewport Editor Header.

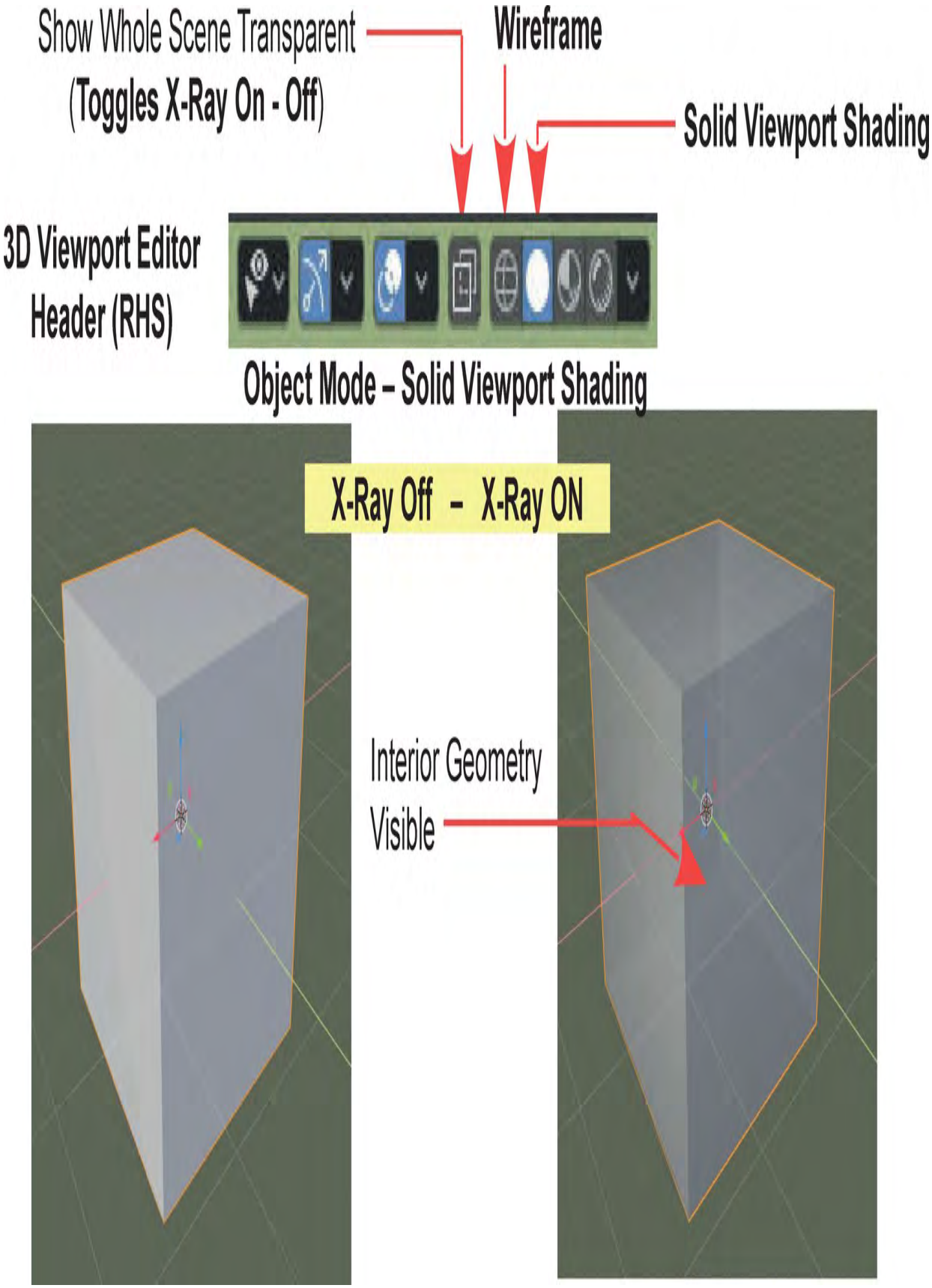

Show Whole Scene Transparent
(Toggles X-Ray On - Off)
Wireframe
Solid Viewport Shading
3D Viewport Editor
Header (RHS)
Object Mode – Solid Viewport Shading
X-Ray Off – X-Ray ON
Interior Geometry
Visible

Figure 5.5

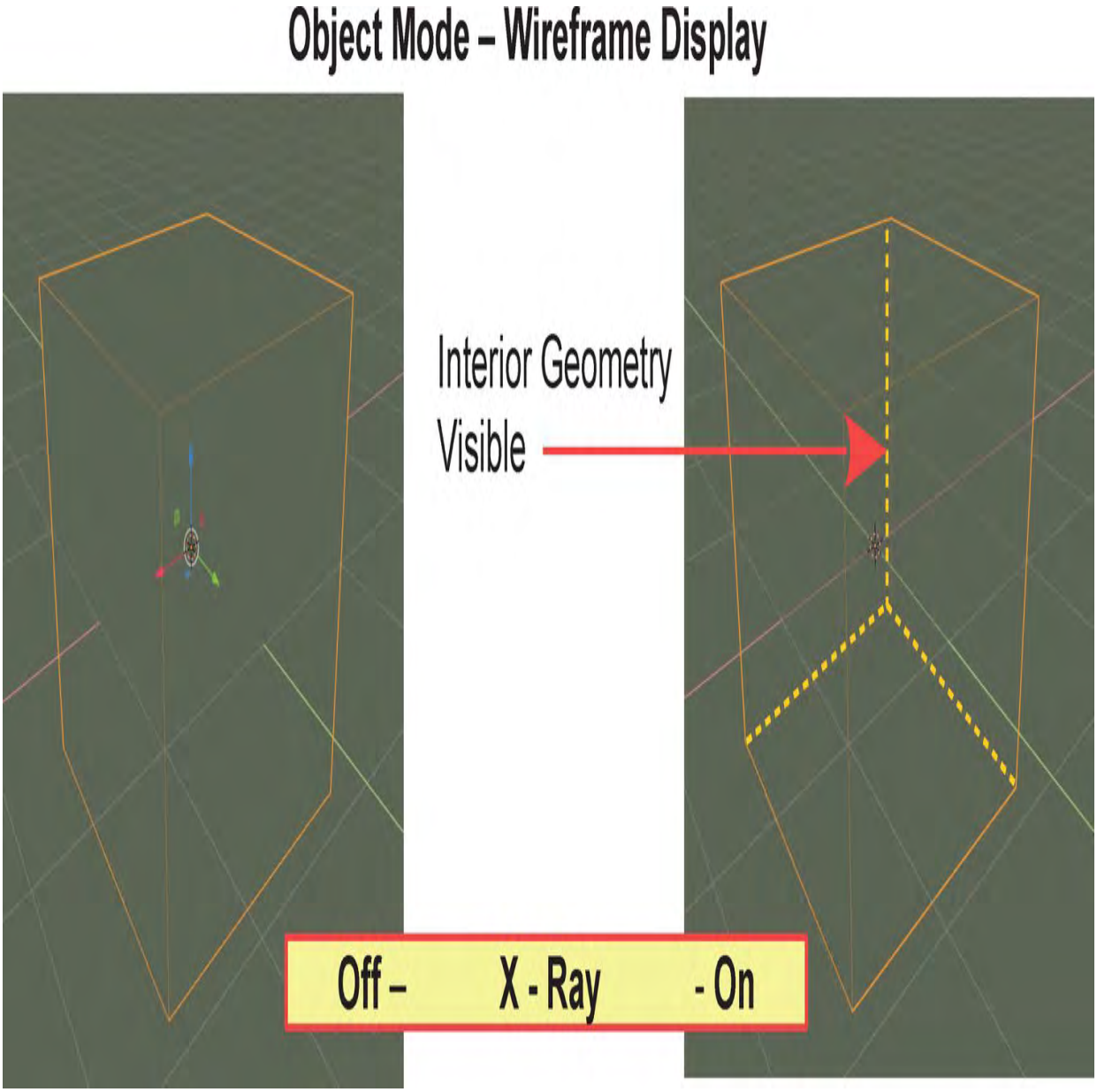


Figure 5.6

5.3 Selecting Vertices, Edges and Faces

While in **Edit Mode**, to select a single Vertex, first press **Click LMB,** in an empty part of the Editor, to deselect all the Vertices. In **Edit Mode** this does not deselect the Object, only the Vertices. Click with the left mouse button

(LMB) on a single Vertex to select it (Vertex displays white). To select multiple Vertices, hold down **Shift** while using the LMB to click. You can also drag a rectangle around the Vertices. Press the **B Key**, hold and drag a rectangle to select a group of Vertices. Pressing the **C Key** will bring up a circle selection tool. Holding the LMB and dragging the circle, selects Vertices on the move. The circle can be sized by scrolling the center Mouse Wheel (MMB). Pressing **Esc** will get you out of the circular selection tool. In order to deselect all Vertices or deselect those currently selected, press the **Alt + A Key** or **LMB** click on an empty space in the **3D Viewport Editor**. The selection and deselection procedures are the same for Vertices, Edges and Faces.

5.4 Manipulating Selected Vertices, Edges and Faces

After selecting Vertices, Edges or Faces, you can use the basic controls used for manipulating Objects (**G Key** to grab or move (Translate), **S Key** to scale, and **R Key** to rotate) (Figure 5.7). Obviously you cannot scale a single vertex but you can scale two or more selected Vertices which constitute a **Vertex Group**. **Note:** The **Edit Mode Tool Panel** at the LHS of the Viewport Editor provides **Manipulation Widgets** for Translating, Scaling and Rotating Chapter 6 - Editing Tools).

Examples of Translation (Move), Scale and Rotation

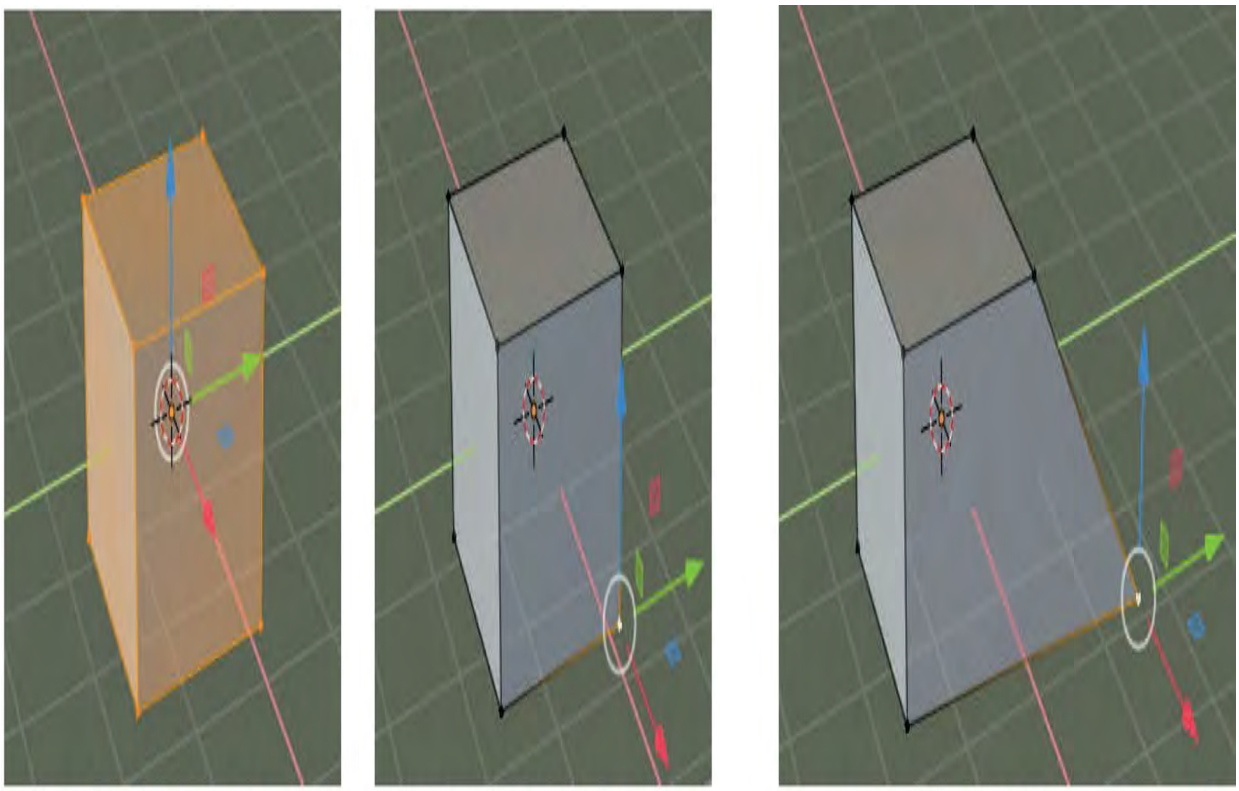
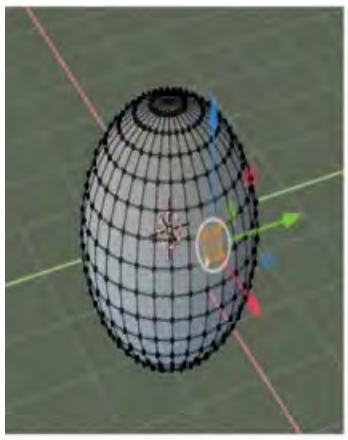
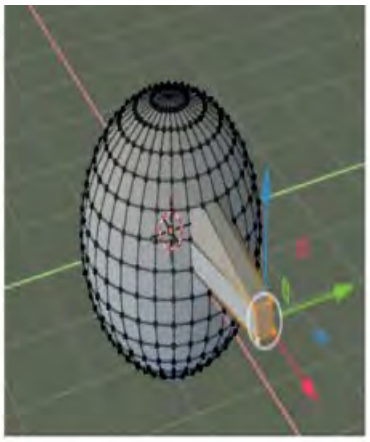

Default Cube – All Vertices Selected
Single Vertex Selected
Single Vertex Moved (Translated)

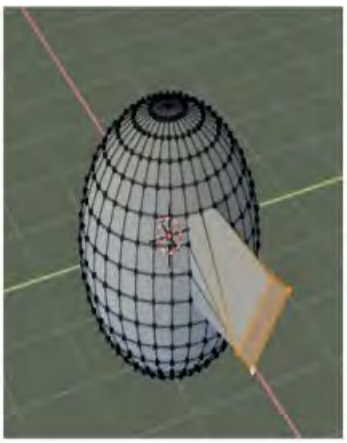
Figure 5.7



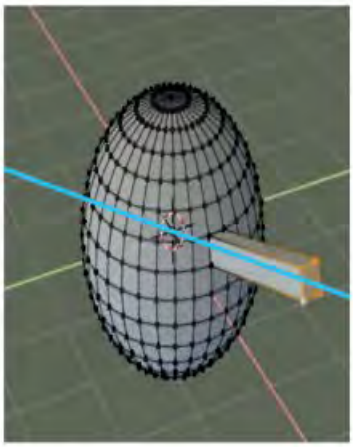
Four Vertices Selected
Hold Shift – Click **LMB**



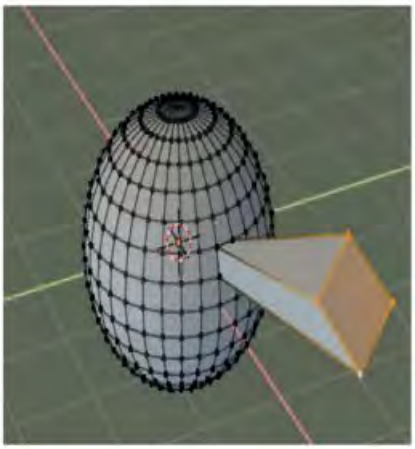
Vertex Group **Translated**
Using the Widget



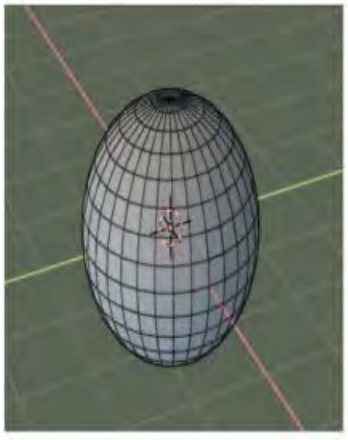
Vertex Group Scaled and Rotated
S Key Scale – R Key Rotate



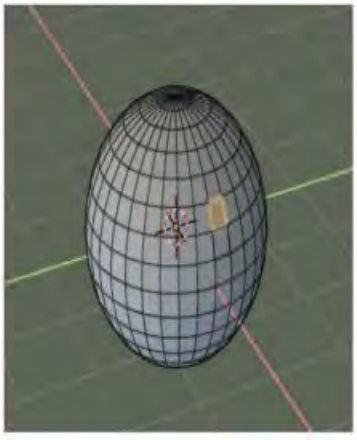
Vertex Group **Extruded** Along
the Axis Normal to the Face



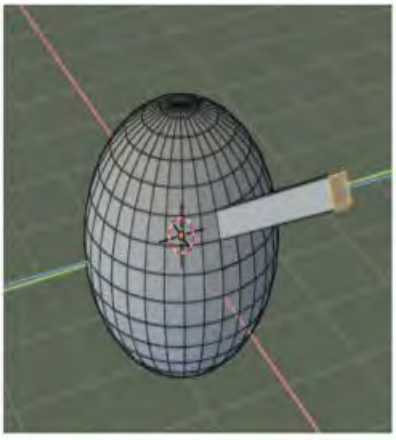
Vertex Group Scaled and Rotated



Face Select Mode



Single Face Selected



Face Extruded Along Axis

Figure 5.8

Note the difference between **Translating (Moving)** a **Vertex Group** and **Extruding** a **Vertex Group**. Extrusion creates new Vertices (see Chapter 6-6.5 Editing Tool – Extrude Region).

5.5 Creating Vertices by Subdivision

There are occasions when you need to add more Vertices to part or **all of the mesh** in order to create detail. To do this, you must be in **Edit Mode** with Vertices selected. To add Vertices to a specific area select Vertices surrounding the area where you wish to add detail. Click **RMB** in the Screen to display the **Vertex Context Menu** and select **Subdivide.**

Figure 5.9

Selecting **Subdivide** inserts a new Vertex at the mid point of each Edge defined by the selected Vertices. Opening the Last Operator **Subdivide Panel** (Lower LHS) shows **Number of Cuts = 1**. Each Edge has been divided once. You may increase the Number of Cuts, further dividing.

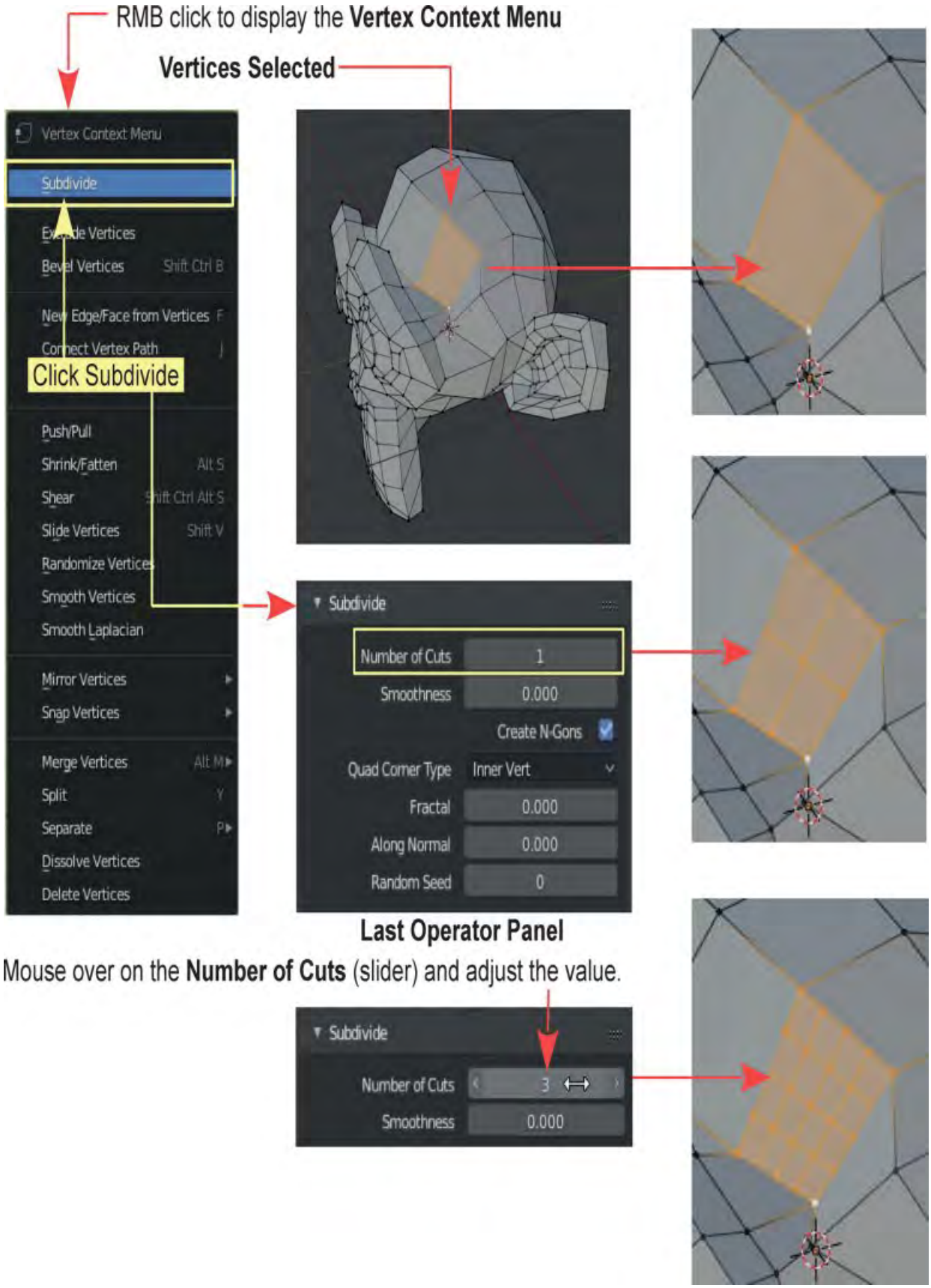

RMB click to display the **Vertex Context Menu**
Vertices Selected
Vertex Context Menu
Subdivide
Bevel Vertices Shift Ctrl B
New Edge/Face from Vertices F
Connect Vertex Path J
Click Subdivide
Push/Pull
Shrink/Fatten Alt S
Shear Shift Ctrl Alt S
Slide Vertices Shift V
Randomize Vertices
Smooth Vertices
Smooth Laplacian
Mirror Vertices
Snap Vertices
Merge Vertices Alt M
Split Y
Separate P
Dissolve Vertices
Delete Vertices
Subdivide
Number of Cuts 1
Smoothness 0.000
Create N-Gons
Quad Corner Type Inner Vert
Fractal 0.000
Along Normal 0.000
Random Seed 0
Last Operator Panel
Mouse over on the **Number of Cuts** (slider) and adjust the value.
Subdivide
Number of Cuts 3
Smoothness 0.000

Figure 5.10

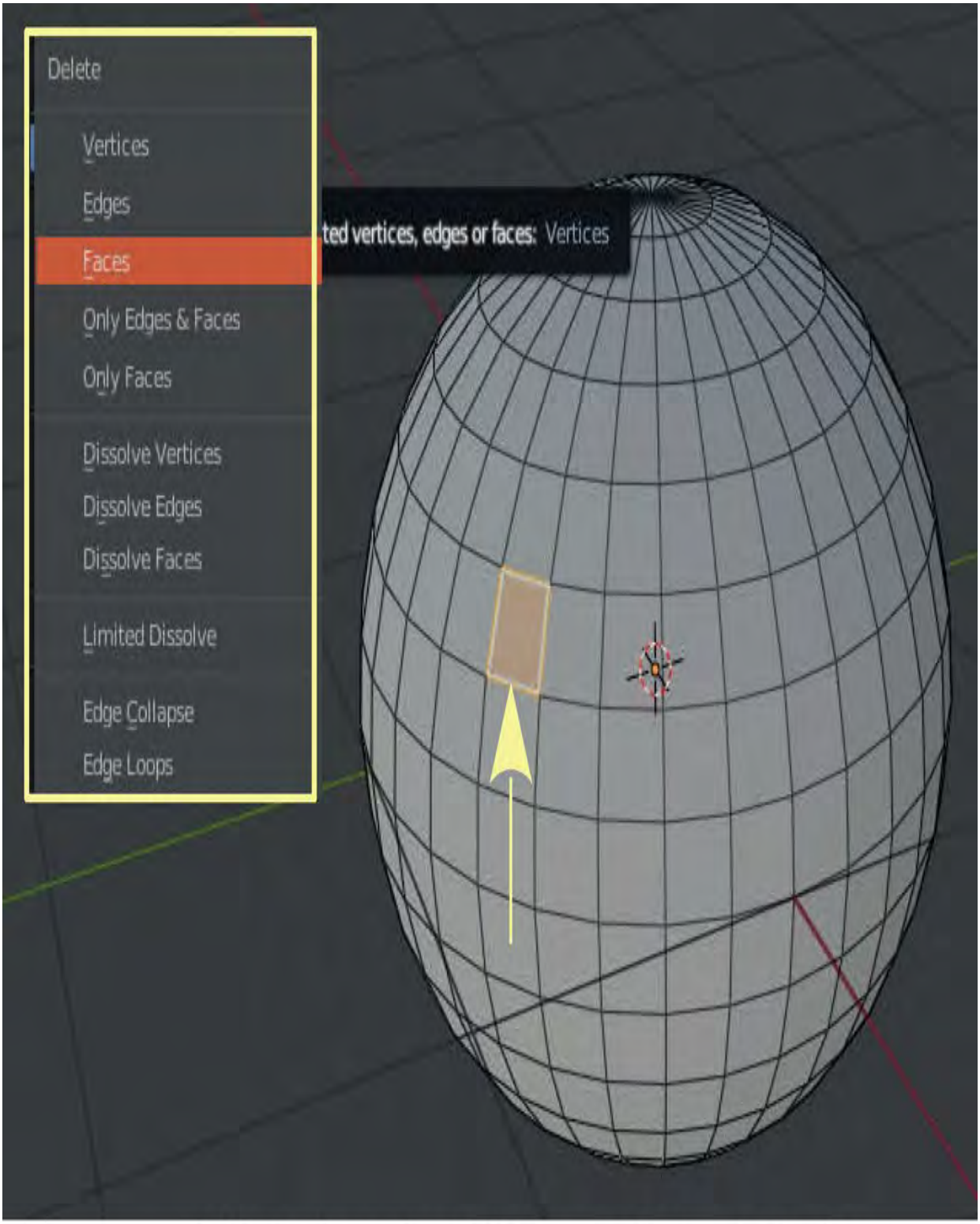
Note: With the initial selection of Vertices the maximum Number of Cuts is 10. To increase beyond 10, have the subdivided area selected, click RMB again and repeat the subdivision process.

Alternative Subdivision Method

Note: With the initial Vertices selected you may also click **Edge** in the 3D Viewport Editor Header and click Subdivide for the same result.

5.6 Adding and Deleting Vertices, Edges, or Faces

Deleting: If you want to make a hole in a mesh, enter **Edit Mode** and select the Vertices, Edges or Faces you wish to remove, then hit the **X Key or Delete Key**. Select (click) the item from the menu that displays (Figure 5.11).

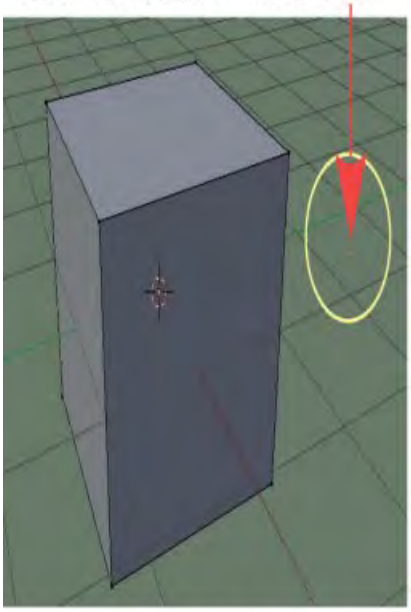


UV Sphere in Edit Face Select Mode

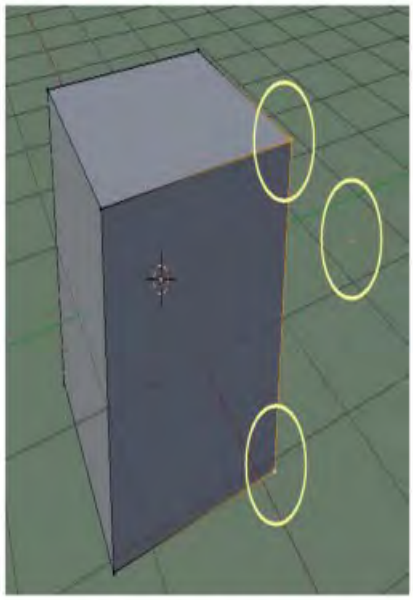
Figure 5.11

Adding Vertices and Faces: Place the Object in **Edit Mode**. Deselect all Vertices. **Press Ctrl** and click **RMB** where you wish to place a new Vertex. Shift select three or more Vertices. Press the **F key** to **Face** (fill in between the selected Vertices Figure 5.12).

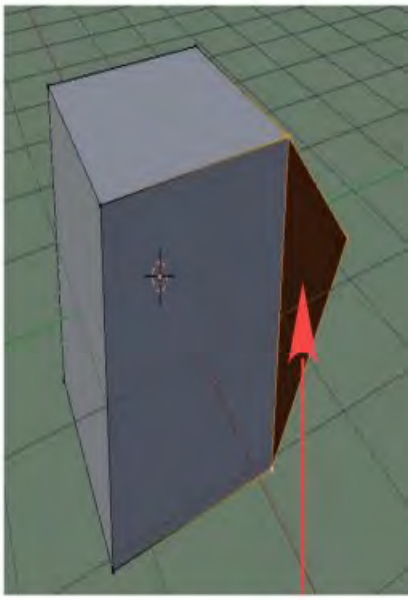
New Vertex Ctrl + RMB click



Cube Object in Edit Mode

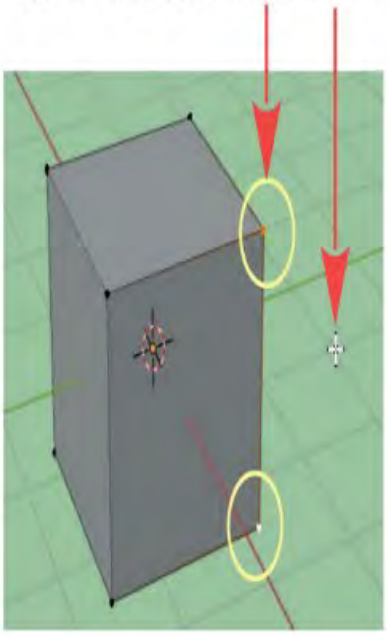


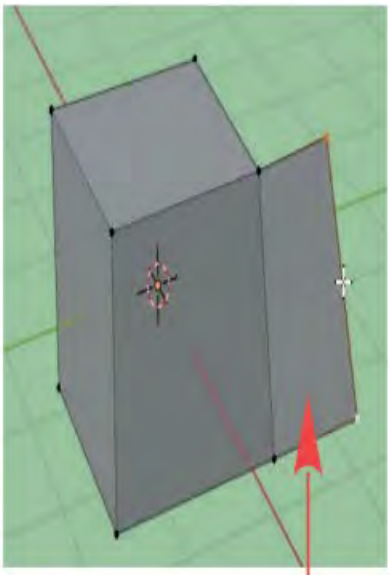
Select Three Vertices



Press **F key** to Face

Two Vertices Selected Mouse positioned to add a Vertex





Ctrl + RMB Clicked

Figure 5.12

Note: If you get the sequence of operations wrong you will get unexpected results.

Adding Faces – User Perspective View

When you add a new Vertex in **User Perspective View** the Vertex is placed on the mid plane of the view. This means it is placed on an imaginary plane located at the center of the 3D World. The plane is normal (at right angles) to your computer screen.

New Vertex added. Three Vertices are selected then Faced (Figure 5.13).

All Vertices are located on the same plane in elevation.

Top Orthographic View

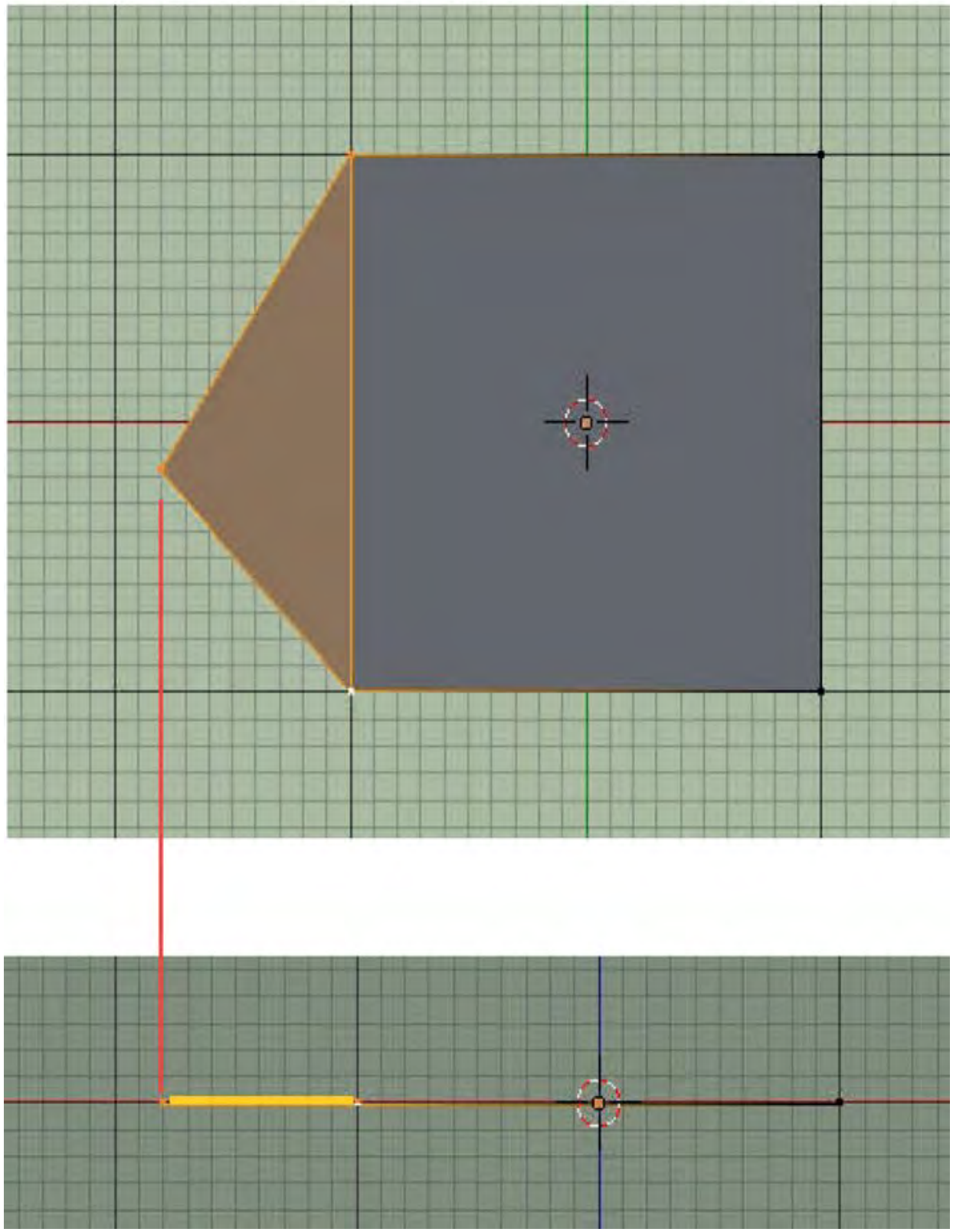


Figure 5.13
Figure 5.14

Front Orthographic View (Figure 5.14)
Side Orthographic View (Figure 5.15)

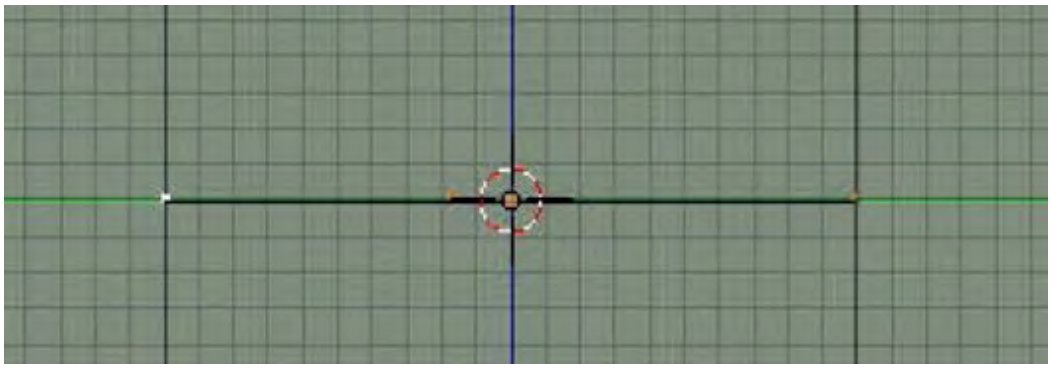


Figure 5.15

New Vertex added. Three Vertices are selected then faced (Figure 5.16).
The new Vertex is displaced in elevation.

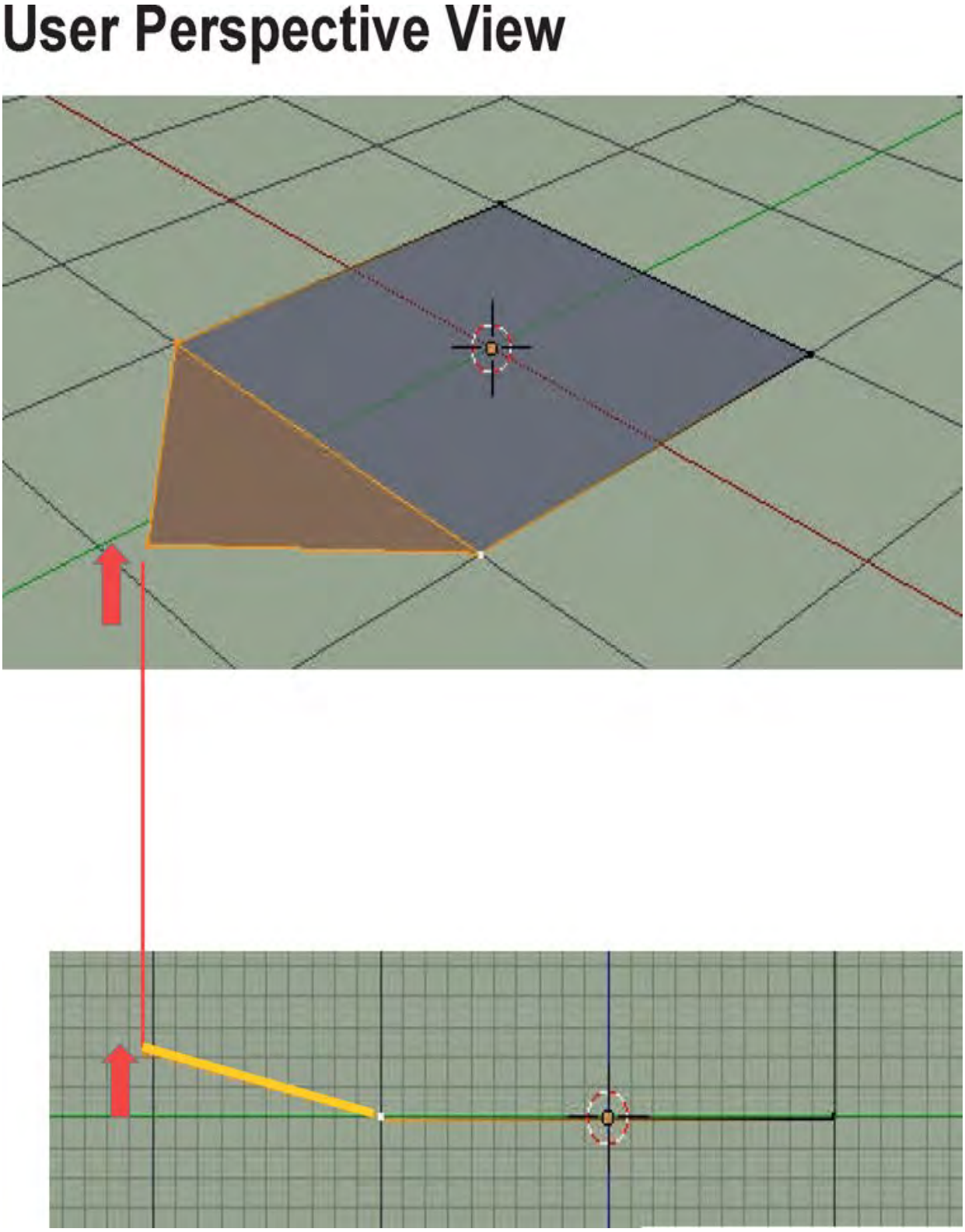


Figure 5.16

Figure 5.17

Front Orthographic View (Figure 5.17).
Side Orthographic View (Figure 5.18).

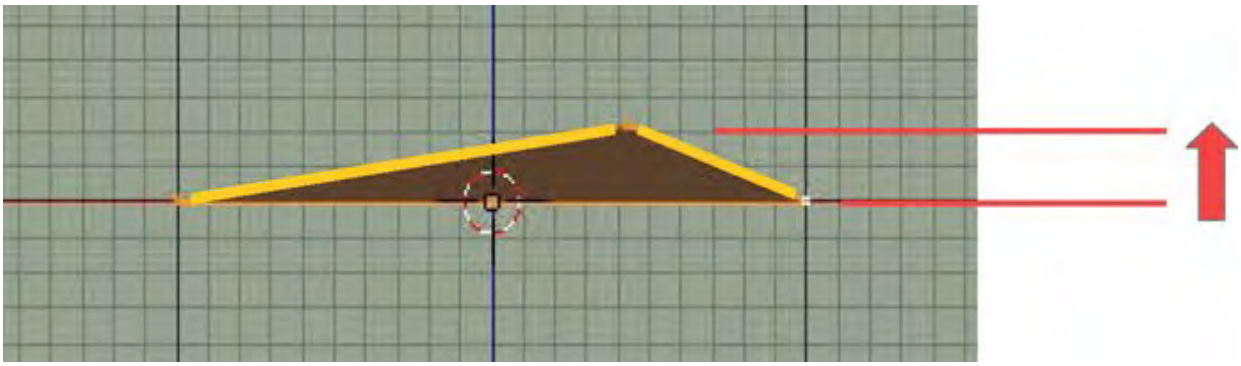
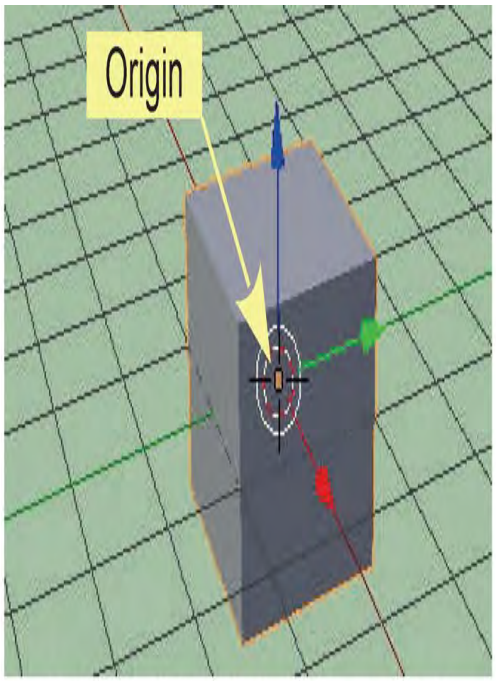


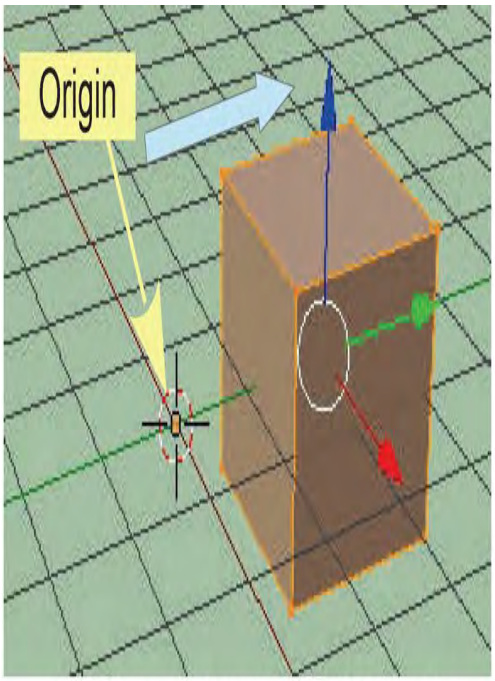
Figure 5.18

5.7 Center Points

Every Object you create in Blender has a small dot somewhere in the center (by default, usually at the center of geometry of the Object). This is the Object's **Origin**, or pivot point (Figure 5.19). With the Object selected in **Object Mode**, moving the Object moves the origin at the same time. In **Edit Mode** the origin does not move when Vertices are moved.

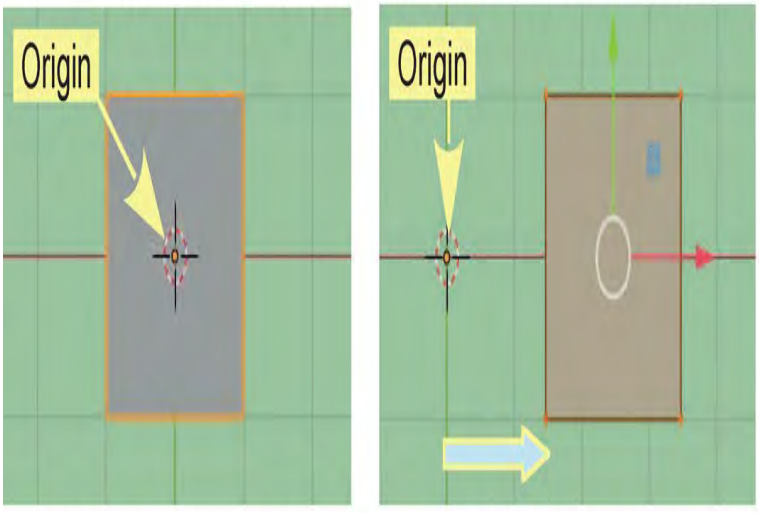


Origin at Center of Geometry

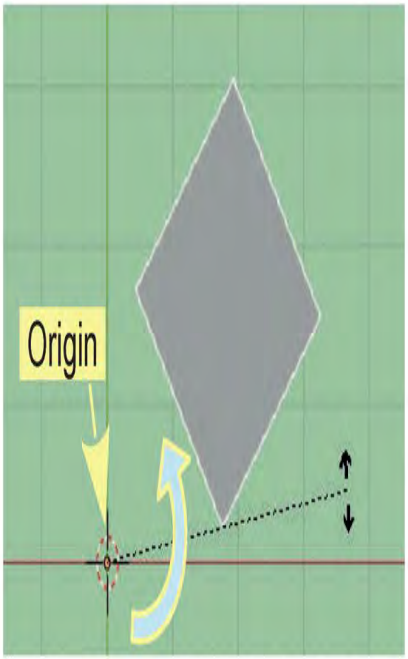


Vertices Moved in Edit Mode

Top Orthographic View (Press Num Pad 7)



In Top Orthographic View – Object Mode – Vertices moved, Rotation is about the original Origin position.



R Key (Rotate)- Drag the Mouse

Figure 5.19

Having the Origin located somewhere else is advantageous when you want to set a Rotation point.

Rotation occurs about the Origin. To rotate about a different position, relocate the Origin to the location of the **3D Viewport Editor Cursor,** when it has been relocates. Press **Shift and RMB click** in the Editor to locate the Cursor. With the Mouse Cursor in the 3D Viewport, **RMB click** to display the **Object Context Menu**. Mouse over on **Set Origin** then click, **Origin to 3D Cursor**. Rotating the Cube will be about the Origin located at the position of the 3D Viewport Cursor.

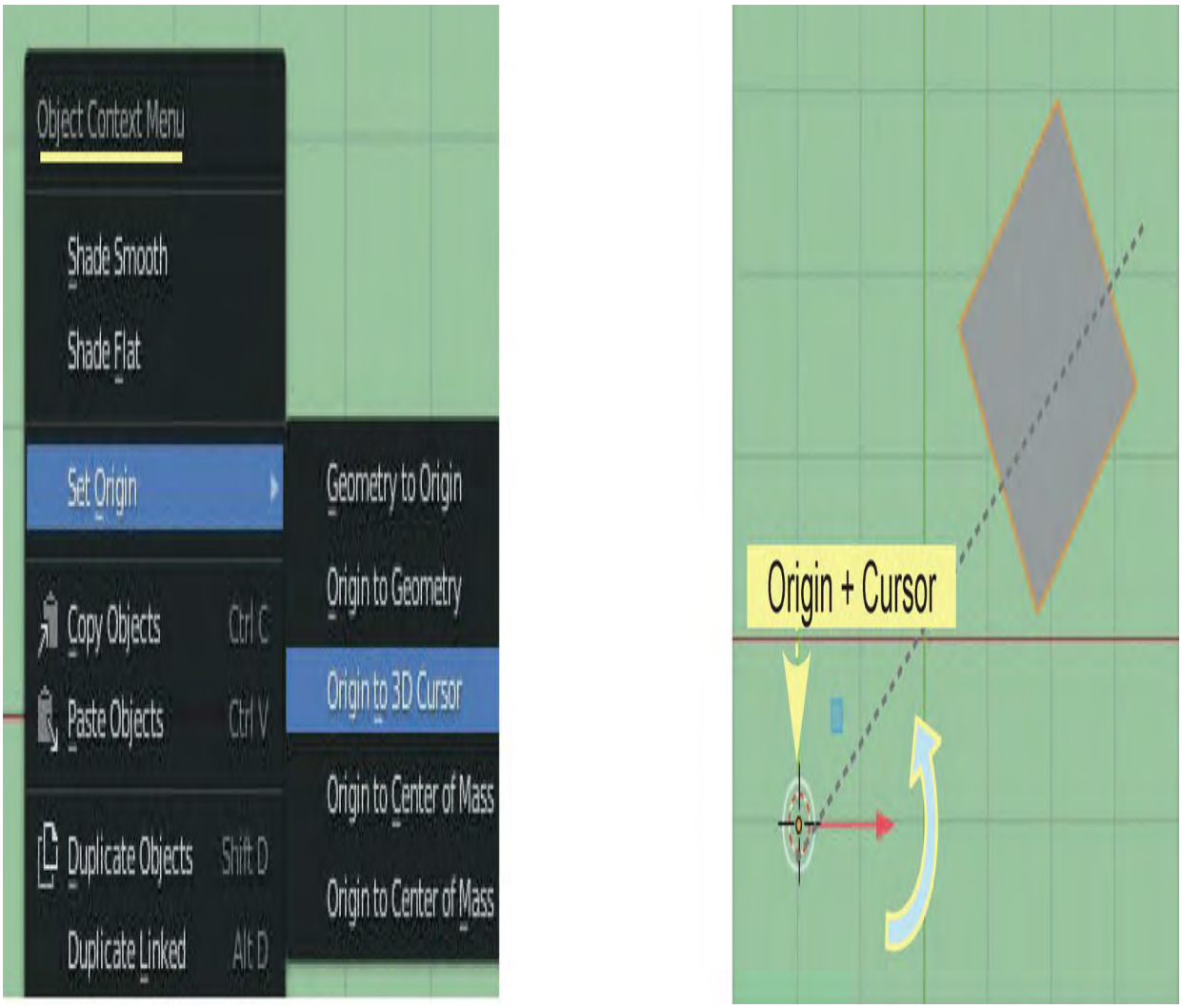


Figure 5.20

In the Object Context Menu you can move the Cube to the new Origin (Geometry to Origin) or move the Origin back to the Cube (Origin to Geometry).

5.8 Joining and Separating

Individual Objects can be joined together to make a single Object. The Objects may be separated under certain conditions or parts of a mesh can be separated from the main Object.

Joining in Object Mode

In Figure 5.21, a UV Sphere Object has been added to the default Scene and positioned such that it intersects with the default Cube. After positioning, the Sphere is deselected and the Cube selected. **Note:** The Move Tool has been selected in the Tool Panel displaying the Translate Manipulation Widget. The Manipulation Widget is located at the center of the Cube.

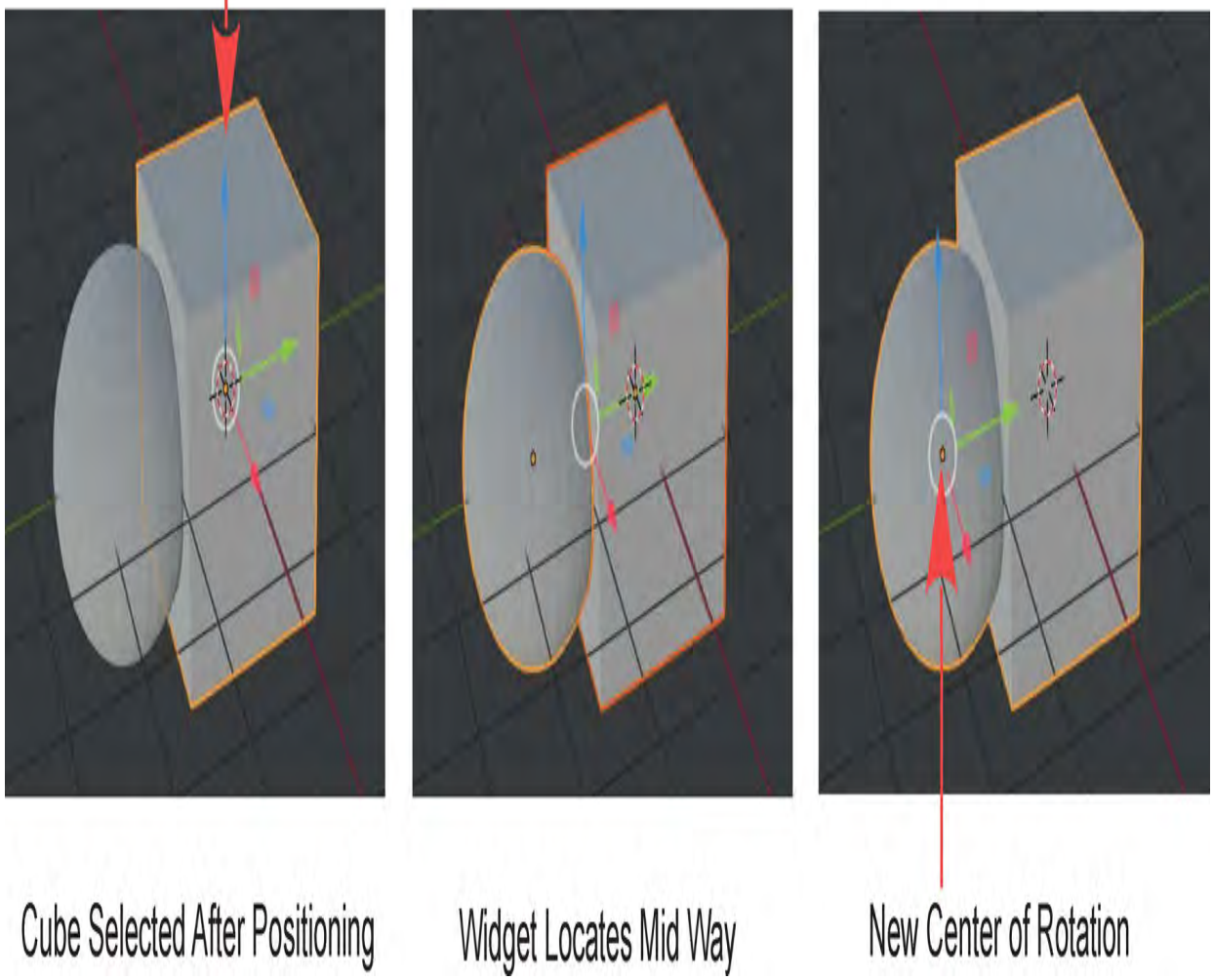


Figure 5.21

To join the two Objects, have the Cube selected and **Shift select the UV Sphere**. Note: The Sphere was the last Object to be selected and now the

Manipulation Widget is located mid way between the center of the Sphere and the Cube. Press **Ctrl + J Key** to join the two Objects. Note: There is an orange outline encapsulating both the Sphere and the Cube which indicates they are a single Object. Note Also: The Widget has located at the center of the Sphere, the last Object selected. The center of geometry for the combined Object is at the center of the Sphere. Rotation of the combined Object will be about this center.

Joining in Edit Mode

Consider the following with the default Blender Scene; The Cube Object is selected in Object Mode. Press **Tab** to enter Edit Mode. Press the **Alt +A Key** deselecting the Vertices. The Cube remains selected since it is the active Object selected in Object Mode. While remaining in Edit mode, add a UV Sphere Object to the Scene. The Sphere will display with its vertices selected but since it coincides with the Cube only the polar and cardinal vertices at the equator are visible.

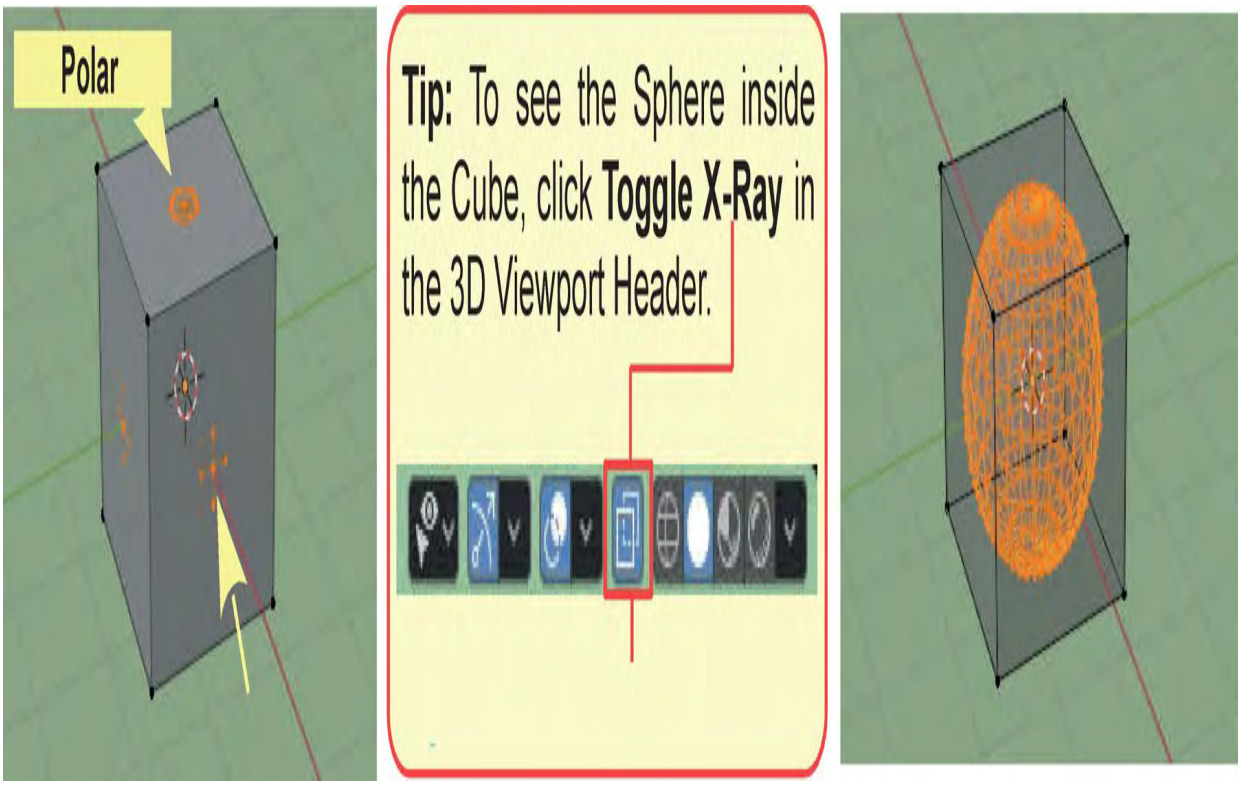
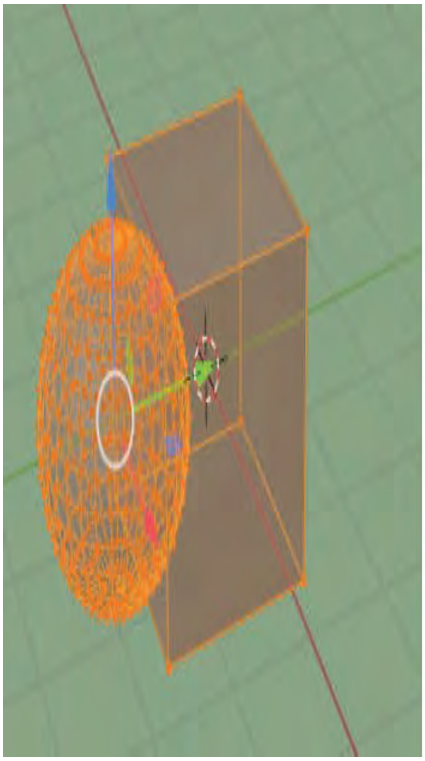


Figure 5.22

Objects added to a Scene in **Edit Mode** are automatically joined to the last object selected while in **Object Mode**.

When the UV Sphere is added all its Vertices are selected and may be manipulated (**T**ranslated **R**otated and **S**caled) manually using the T, R and S Keys or by using the manipulation widgets from the Tool Panel. When manipulated press **Alt** + **A** Key or click LMB in an empty part of the Editor to deselect the Sphere's Vertices. Pressing the **A Key** following this will select the joined UV Sphere plus the Cube. They have been joined and form a single Object (Figure 5.23).



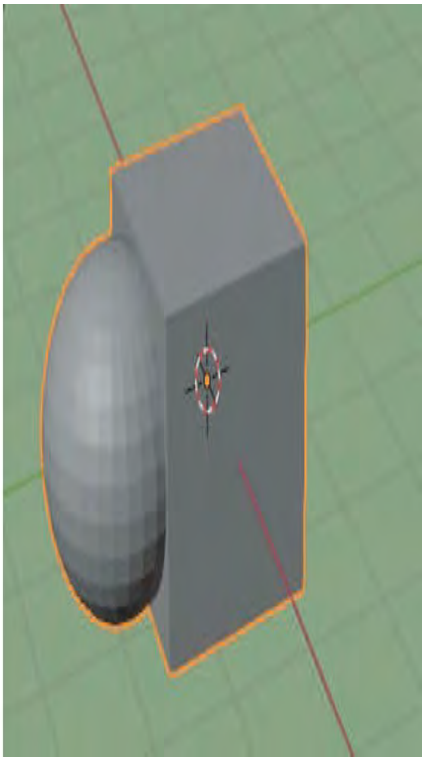


Figure 5.23

In the **Outliner Editor** you will see a single Object named **Cube** listed (the combined Object).

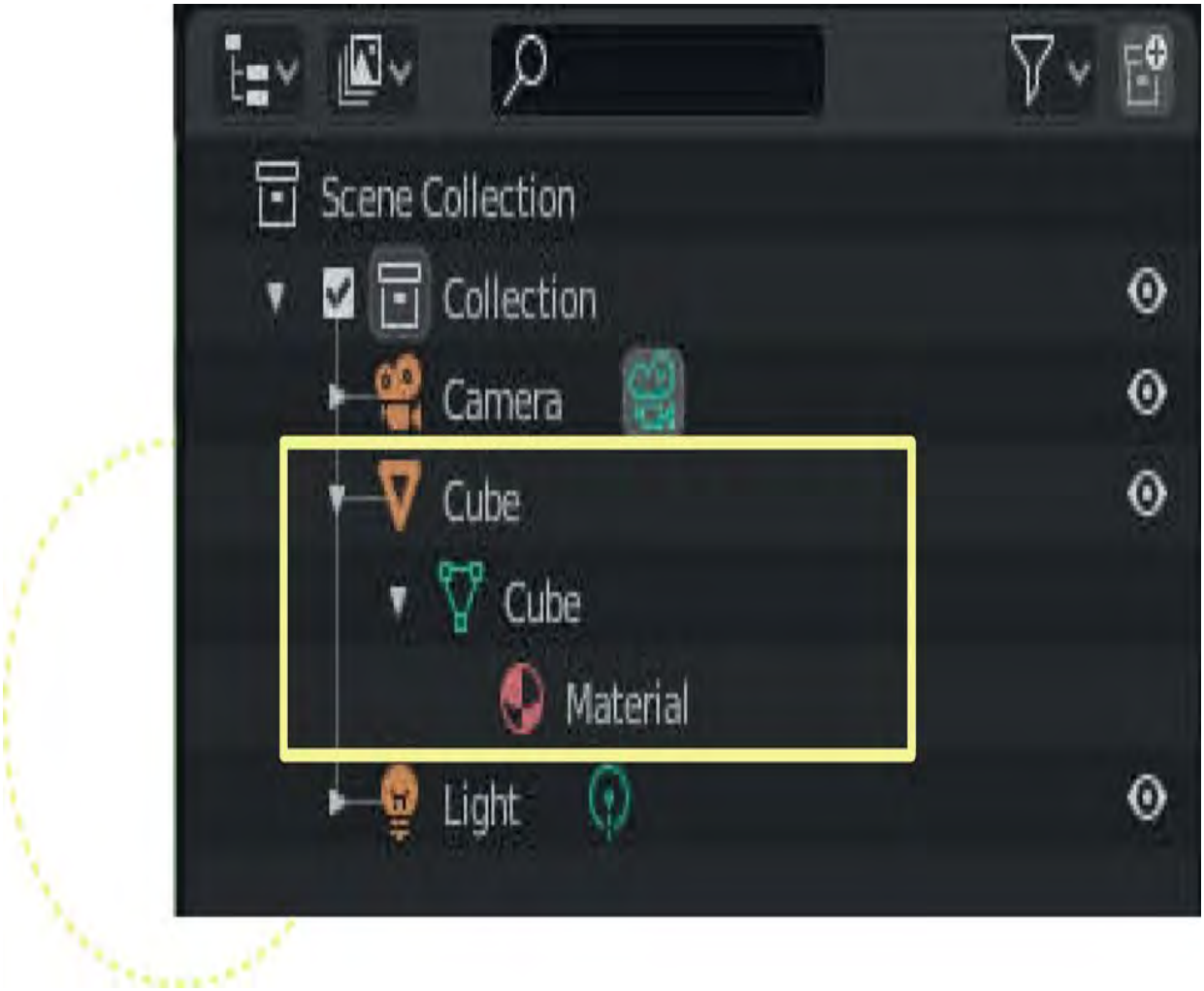


Figure 5.24

Separating in Edit Mode

To demonstrate separation in Edit Mode, use the same Cube – Sphere arrangement as shown in Figure 5.23. Have the Cube and the Sphere joined into a single Object.

Place the **3D View Editor** in **Right Orthographic view** and with the combined Object selected, **Tab** to Edit Mode. Press **Alt + A key** to deselect all Vertices. Press the **B key** (Bo Select) and drag a rectangle around the LH side of the spherical part. **Don't forget to Toggle X Ray On** to enable selection of Vertices behind the visible Vertices.

The selected Vertices will be displayed orange. (Figure 5.25)

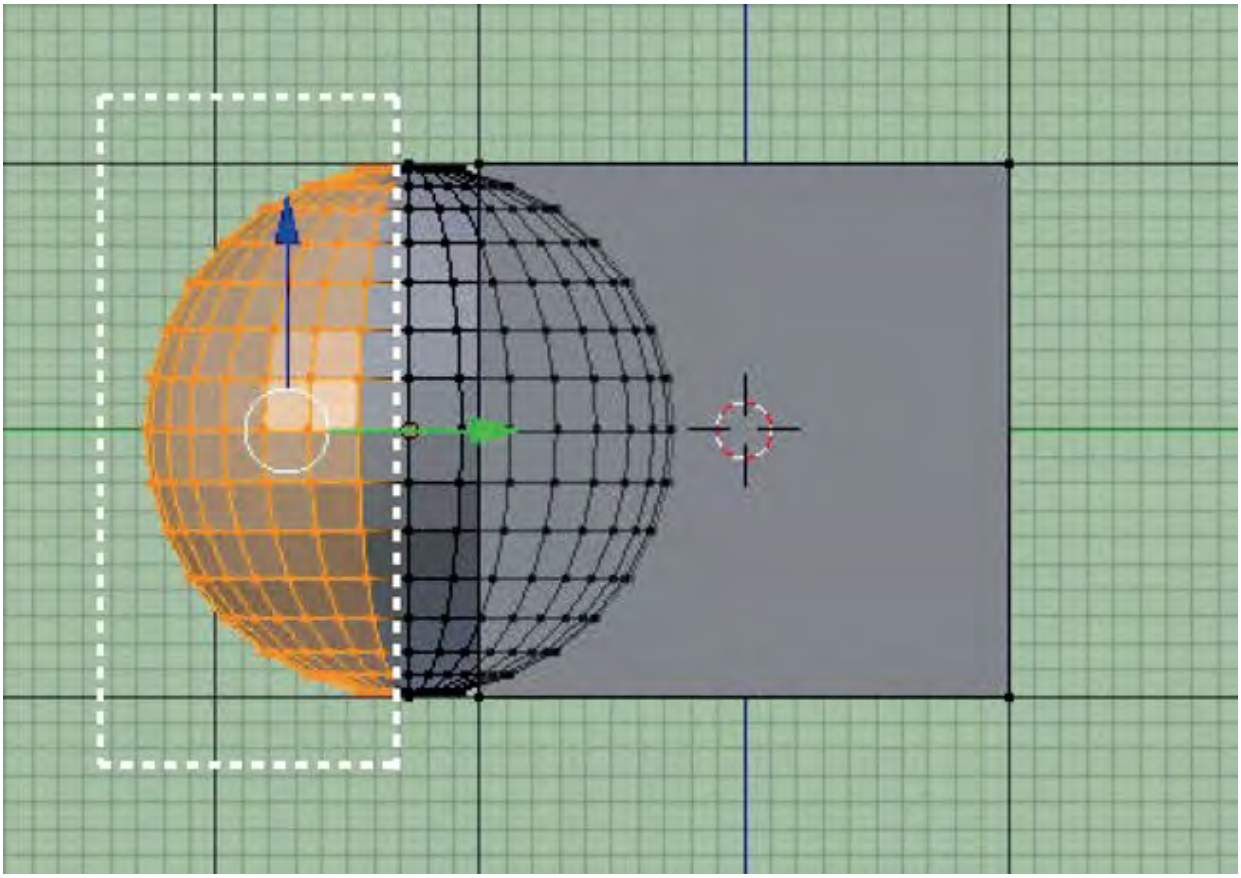


Figure 5.25

With the selection made, press the **P Key** to display the **Separate Menu** and in this case, click on the **Selection** option. The **Selection** option separates the selected Vertices creating a new Object as shown by the orange outline. (Figure 5.26).

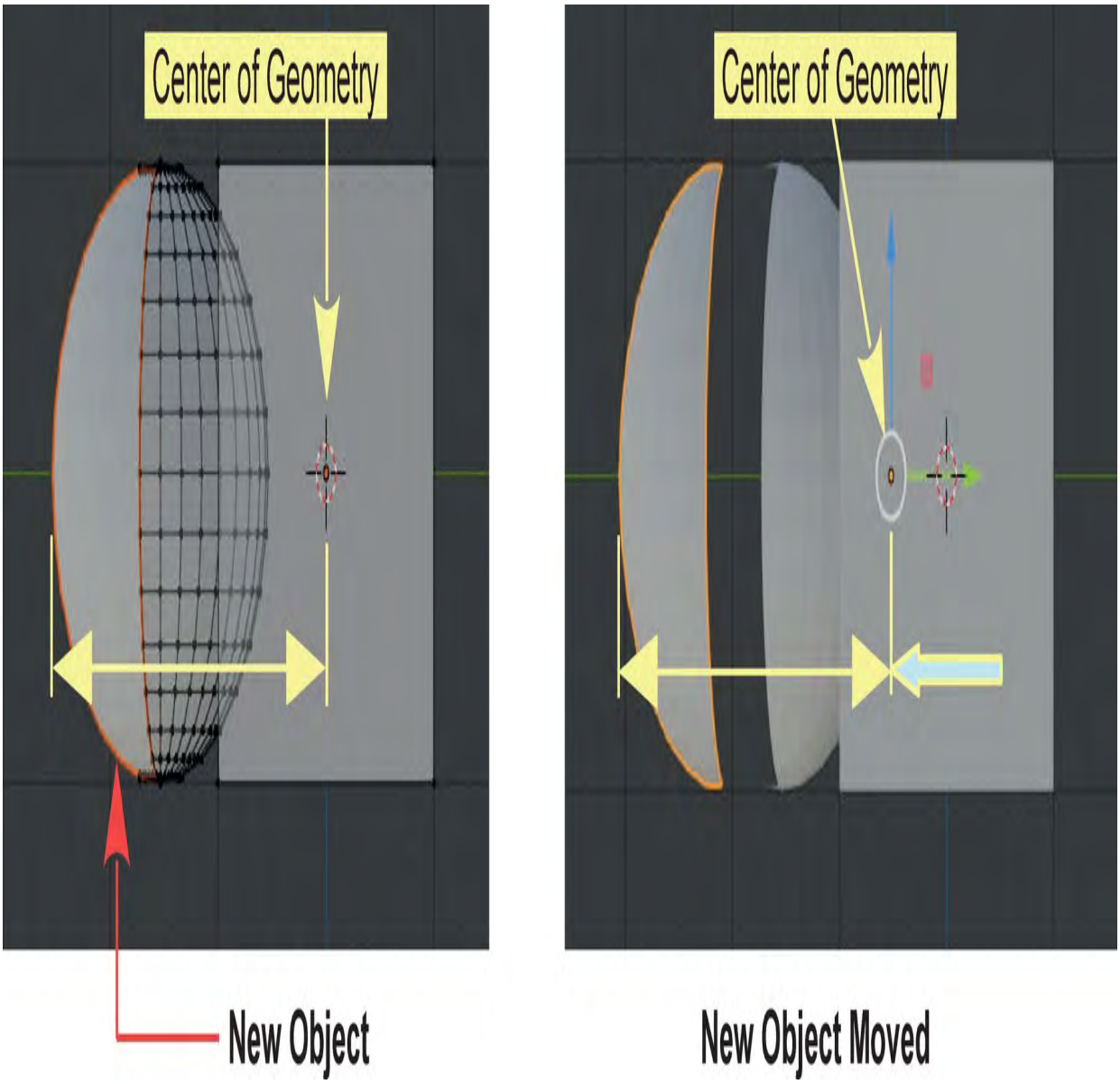


Figure 5.26

Tab back to Object mode. With the Manipulation Widget turned on, move the new Object along the Y Axis. You can move the object anywhere you like, rotate it, scale it etc.

Note: In joining the Sphere to the Cube in Object Mode by pressing **Ctrl + J**, in this case the Cube was the last Object selected, therefore, the center of geometry of the combined Objects was at the center of the Cube. After separating part of the Sphere mesh, its' center of geometry is also located,

coinciding with the center of the Cube. When moving the separated part of the Sphere you see the center of geometry, i.e. center of rotation, displaced from the mesh.

The Separate Menu Options

In the previous example you used the **Selection option** in the **Separate** menu. There are two alternatives (Figure 5.27):

Figure 5.27

Separate by Material: When joined objects have different Materials (Color) applied they will be separated into single Objects according to their color.

> Material (Color) is discussed in Chapter 16

Separate by Loose Parts: Consider the Objects automatically joined in Edit Mode (Figure 5.23).

With the UV Sphere and Cube joined in Edit Mode, and displayed in Edit Mode, press the **P Key** to display the **Separate** menu and select **By Loose Parts**. Immediately one of the Objects will display a red outline. **Tab to Object Mode**. You may now select either part as a separate Object. Note the center of Geometry (Origin).

5.9 Creating Vertex Groups

On occasion you will want to manipulate a **group of vertices**. You can select multiple vertices on an Object and manipulate them, but once deselected you could have trouble selecting the exact same group the next time. To assist with this you can assign multiple Vertices to a designated group for re-selection. Working through the following example will show you how.Figure 5.28

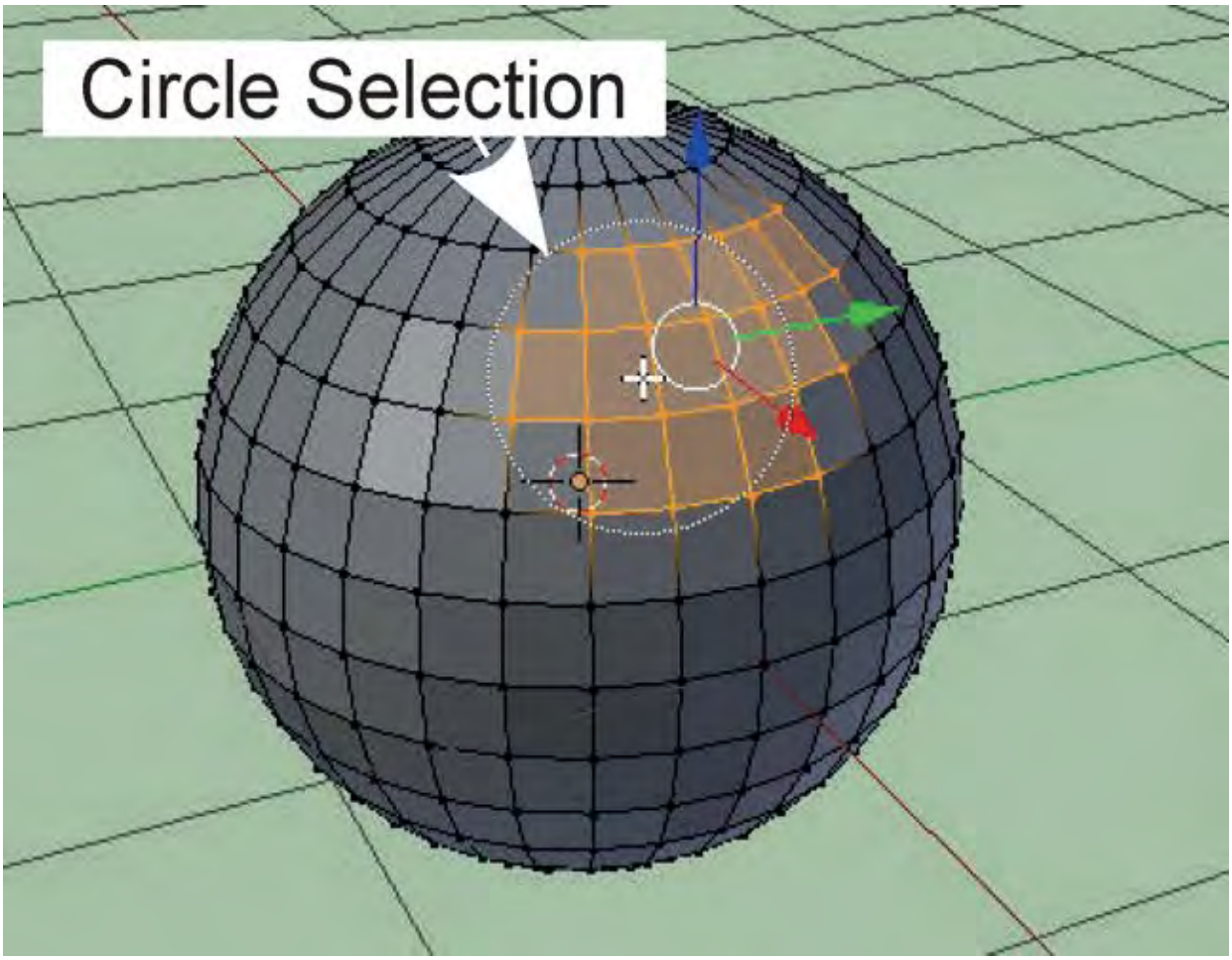


Figure 5.28

Start the default Scene and replace the Cube with a **UV Sphere**. Zoom in on the Scene to give a better view (press the Number Pad + sign). Tab to **Edit mode** and then press

Alt + A key to **deselect** all the Vertices. Press the **C key** for circle select (scroll the mouse wheel to adjust the circle size) and click, hold and drag the circle over the Sphere to select a group of Vertices (Figure 5.28). Press **Esc** to cancel the circle selection. The Vertices remain selected.

In the **Properties Editor, Object Data Properties, Vertex Groups tab**, click on the + sign to create a **Vertex Group Data Slot**. By default, this will be named simply **Group** (Figure 5.29). If you wish you can change the name

to something meaningful by clicking on **Group** in the **Name** slot, deleting it, and retyping a new name.

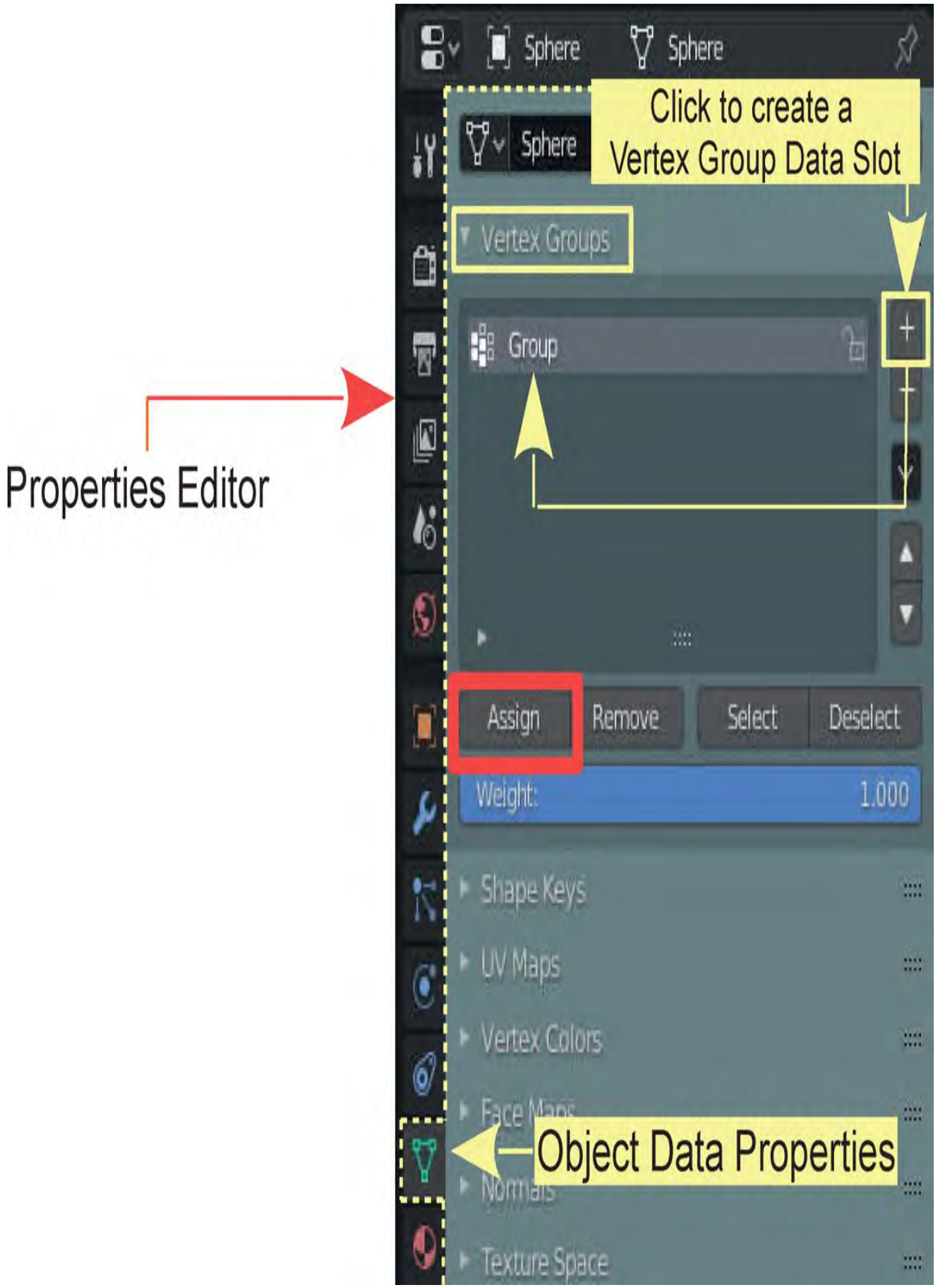

Properties Editor
Sphere
Sphere
Sphere
Click to create a
Vertex Group Data Slot
Vertex Groups
Group
Assign
Remove
Select
Deselect
Weight:
1.000
Shape Keys
UV Maps
Vertex Colors
Face Maps
Normals
Texture Space
Object Data Properties

Figure 5.29

With the group of vertices still selected on the Sphere, click on the **Assign button** in the **Vertex Groups Tab -** this assigns the selected vertices to **Group**.

By clicking on the **Select** and **Deselect** buttons, you will see the vertices on the Sphere being selected or deselected, respectively.

Deselect the Vertices and repeat the circle select with a different group. Click on the + sign again in the **Vertex Groups Tab** and you should see a new data block created named **Group.001**. Click the **Assign button** to assign the new group of vertices to **Group.001**. Deselect the Vertices on the Sphere in the 3D Viewport Editor, and you can now select **Group** or **Group.001** (Figure 5.30).

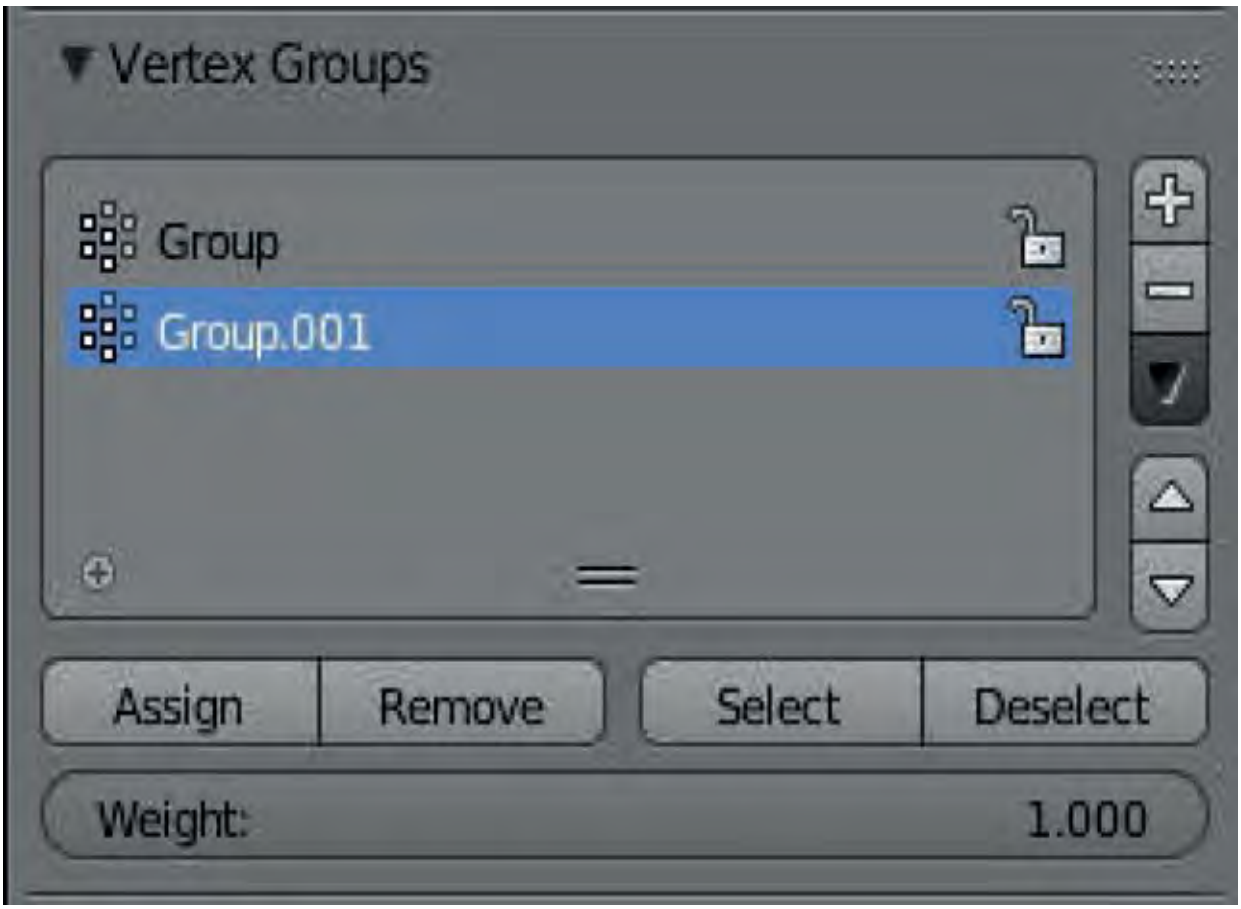


Figure 5.30

Weigh Paint Method

An alternative method for selecting Vertices for assignment to a Vertex Group is to use Blender's **Weigh Paint Tool.**

Place the 3D Viewport Editor in **Weigh Paint Mode** (Figure 5.31).

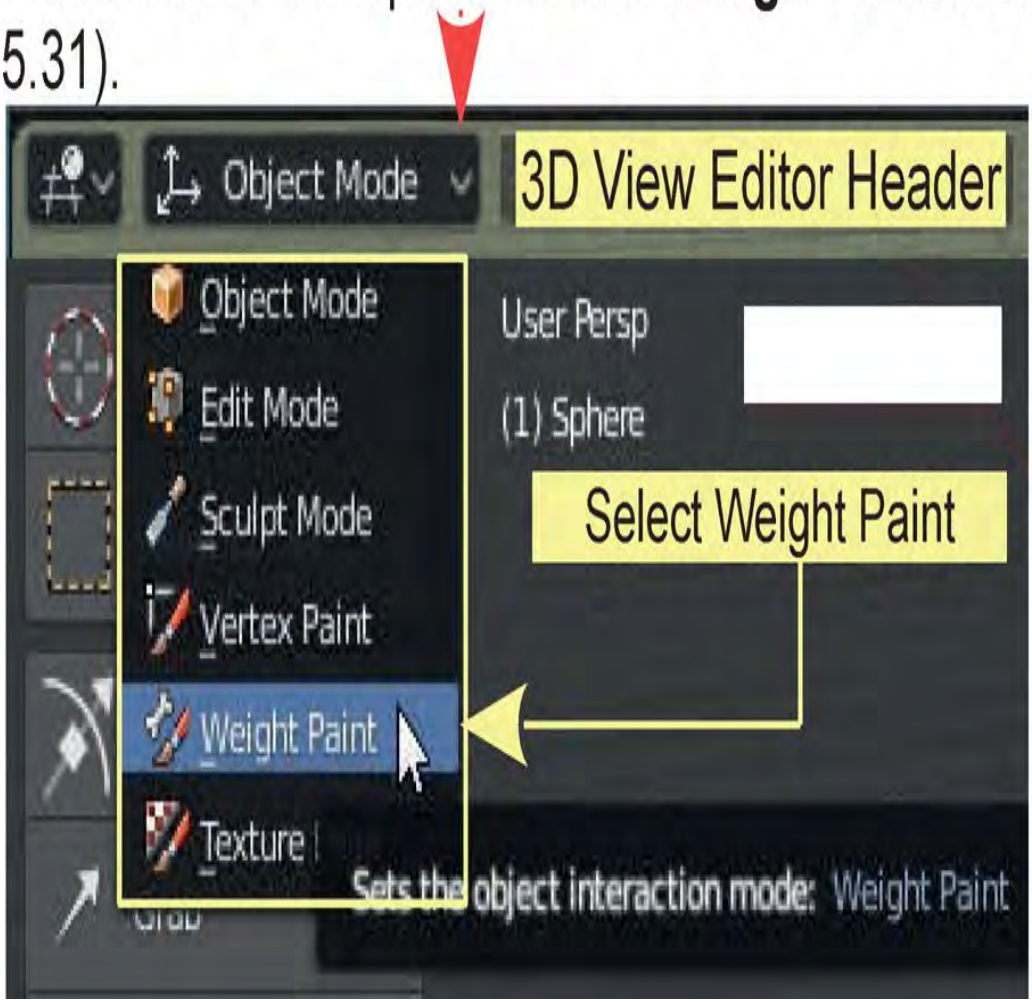


Figure 5.31

In the **Weigh Paint Mode, Tool Panel** click on **Draw.**

The 3D Editor cursor becomes a red circle which is the paint **Brush** (Figure 5.32).

With a **UV Sphere** selected in the **3D Viewport**, go to the **Properties Editor, Object Data Properties**. In the **Vertex Group Tab** click the **Plus** sign to create a new **Vertex Group**. The Vertex Group will be named simply **Group** (Figure 5.32).

Figure 5.32

Figure 5.33

The UV Sphere displays blue. Click, hold and drag the brush over the surface of the UV Sphere painting an area until it is red (Figures 5.34, 5.35). Painting selects Vertices.

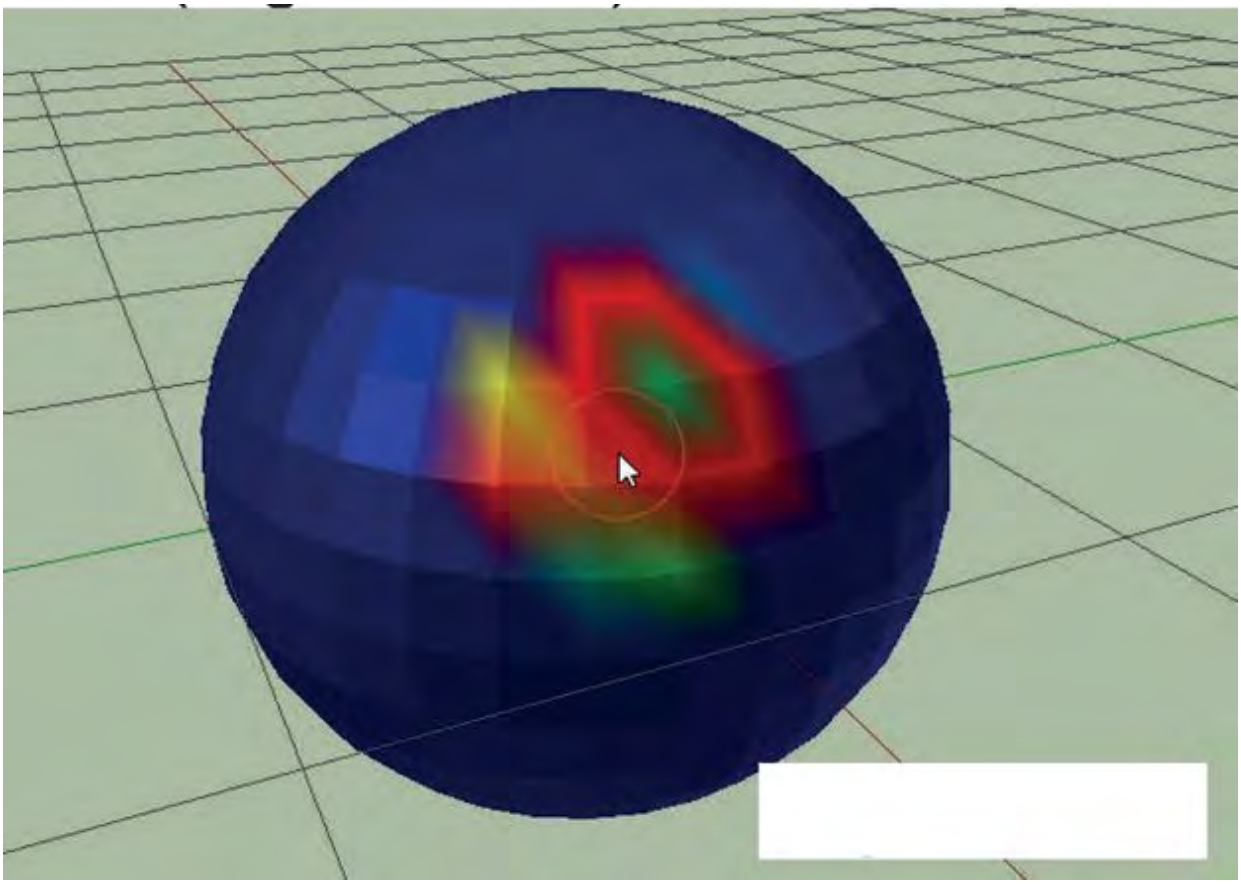


Figure 5.34

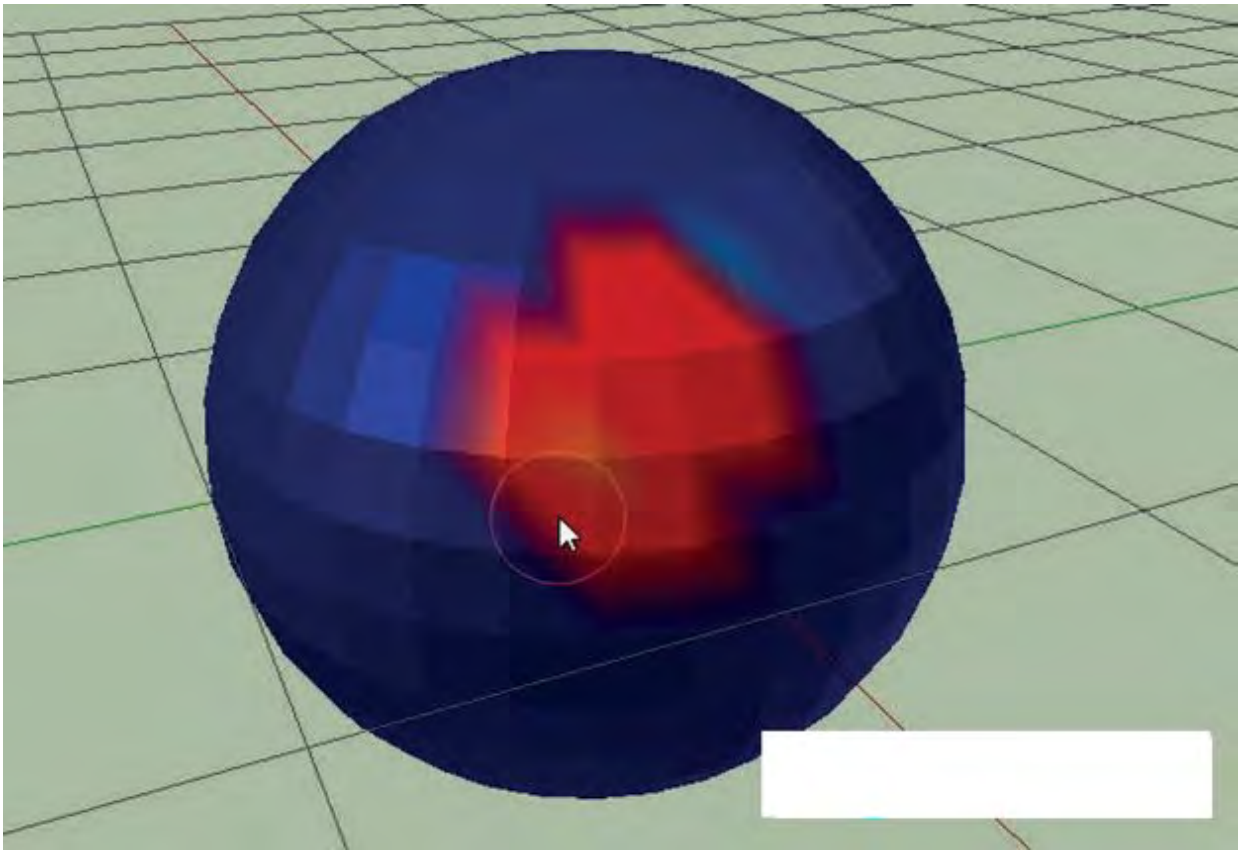


Figure 5.35

Press the **Tab Key** to place the 3D Viewport in **Edit Mode** and **deselect all Vertices** (press Alt + A Key).

In the **Properties Editor, Object Data Properties, Vertex Groups Tab,** with the new Vertex Group named Group highlighted, click the **Assign button** (Figure 5.36).

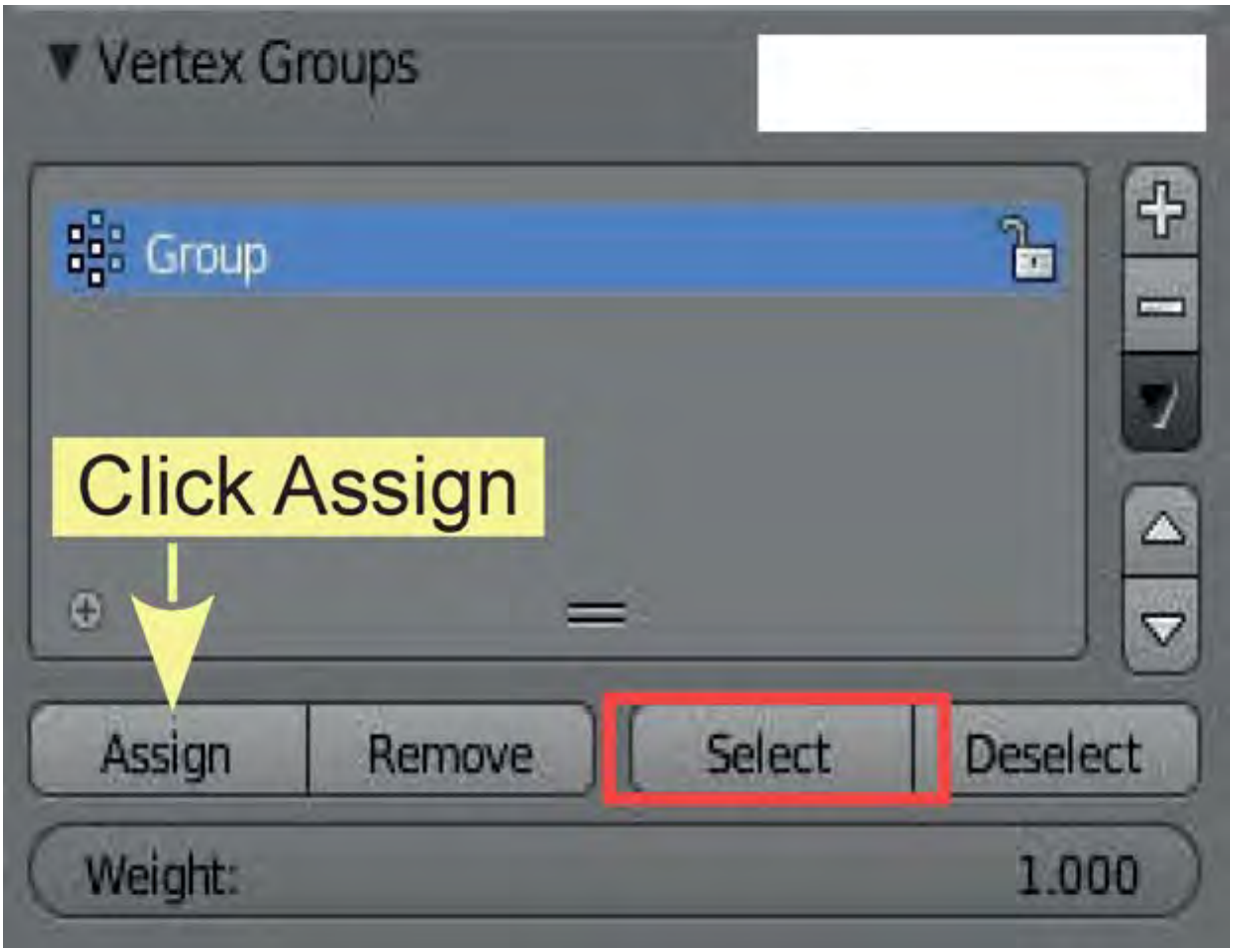


Figure 5.36

Click the **Select button** to see the painted vertices assigned to the **Vertex Group** (Figure 5.37).

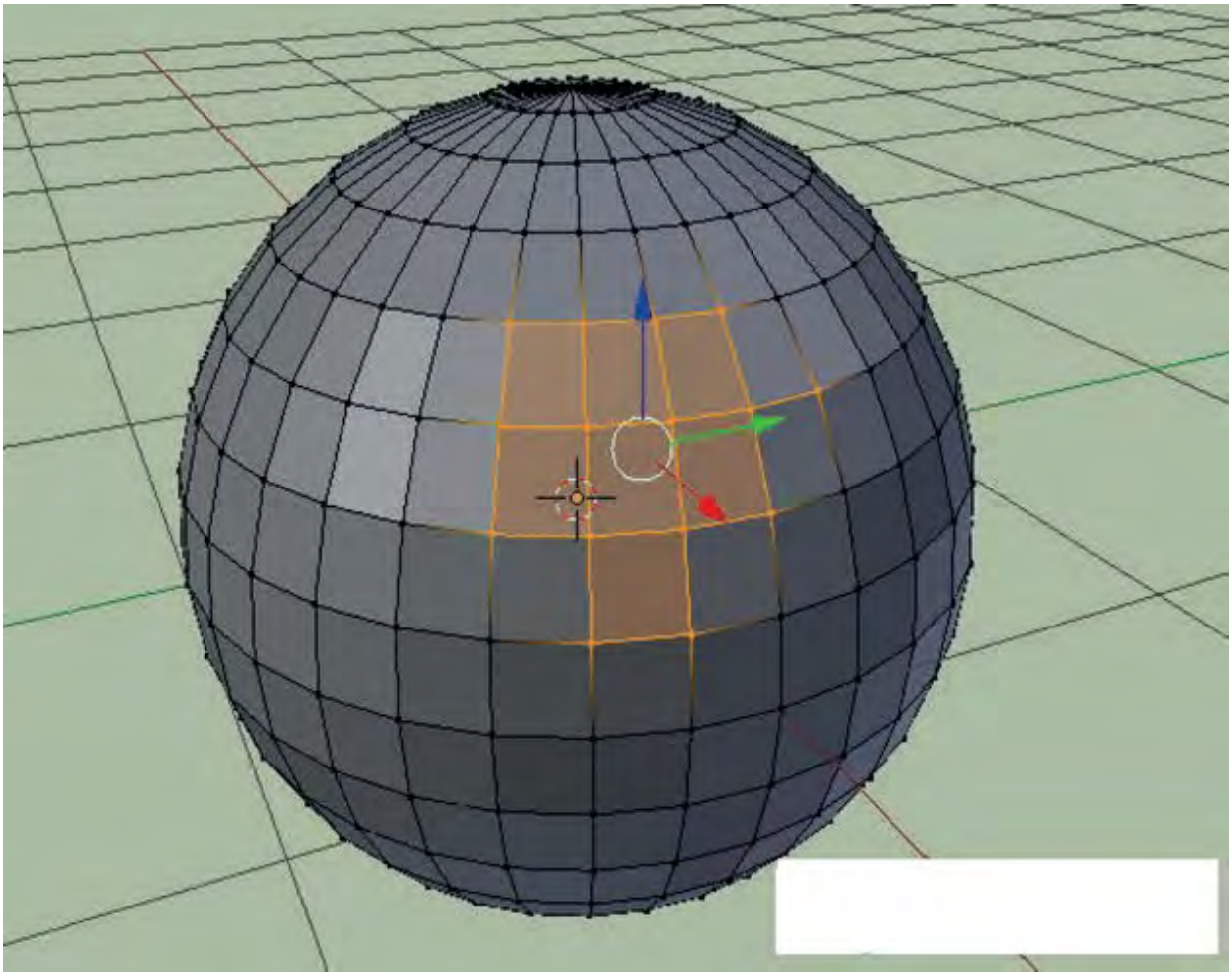


Figure 5.37

5.10 Proportional Vertex Editing

Proportional Vertex Editing is used to create a flow in the shape when editing Vertices. To turn **Proportional Vertex Editing on**, in **Edit Mode**, click the **Proportional Editing** button in the **3D View Editor Header** (Figure 5.38).

Zoom in on a Plane Object in **Edit Mode** in the 3D Viewport. In Edit Mode, subdivide the Plane (**RMB click – Subdivide**) with Number of Cuts = 10. Deselect the Vertices then **re-select** (press **A Key**)and repeat the subdivision with Number of Cuts = 2.

Figure 5.38

With Proportional Vertex Editing enabled there are several types of Falloff available. To see what this means perform the following:

Note: To adjust the **Circle of Influence** select a Vertex or Vertices. Press the **G Key** (grab) and scroll the Mouse Wheel. (See over page)

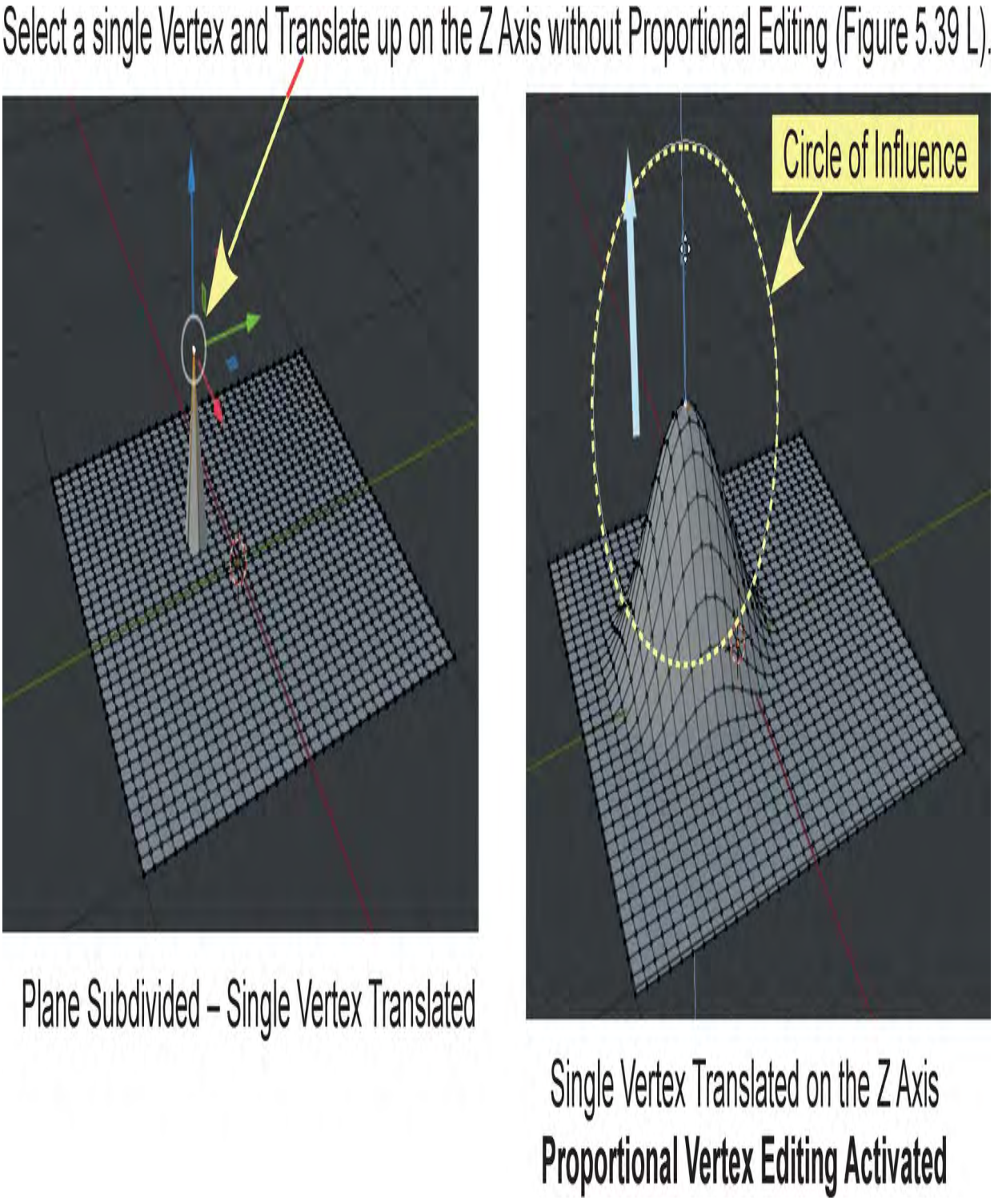


Figure 5.39

Circle of Influence

When a single Vertex is selected and Translated **with Proportional Vertex Editing activated** a **Circle of Influence** determines how many surrounding Vertices are influenced in the Translation.

The diameter of the circle is adjustable. Select the Vertex to be Translated. Press the G Key (places the Vertex in Grab Mode) to display the circle. Without moving the Mouse, scroll the mouse wheel. Click LMB. The circle display is cancelled but the influence is retained. Pressing the G Key and moving the Mouse displays the circle again and moves the selected Vertex. Clicking the Move button in the Tool Panel displays the Manipulation Widget and allows Translation of the Vertex. The Circle of Influence previously set is in effect. Experiment with the different types of Falloff.

Having moved the Vertex, the Last Operator – **Move Panel** displays which includes a Proportional Size slider for adjusting the Circle of Influence.

It doesn't take much imagination to see that Proportional Vertex Editing as seen in Figure 5.40 can be employed to create a landscape (see Chapter 11 – 11.1 Editing Techniques and Examples).

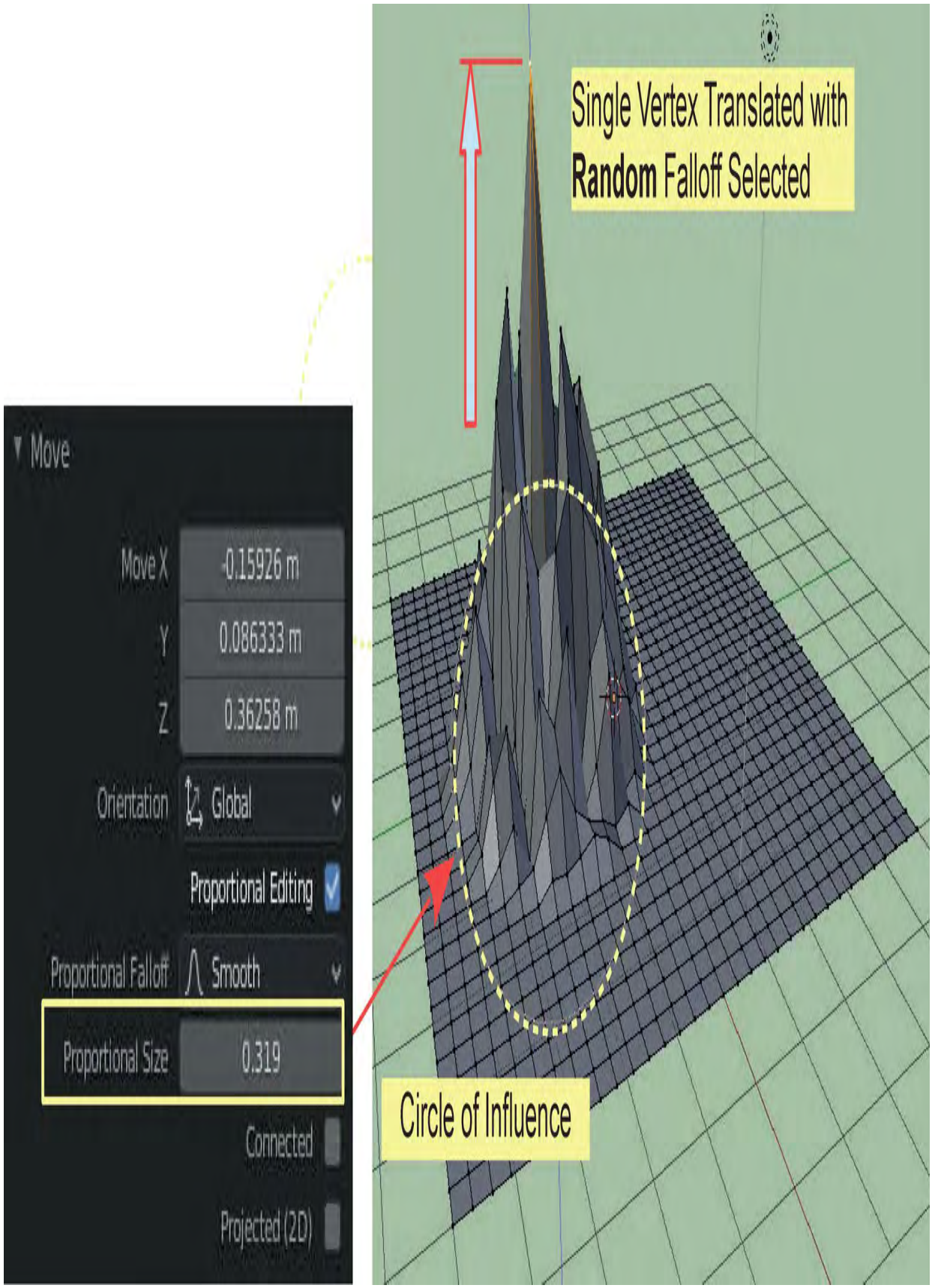

Move
Move X
-0.15926 m
Y
0.086333 m
Z
0.36258 m
Orientation
Global
Proportional Editing
Proportional Falloff
Smooth
Proportional Size
0.319
Connected
Projected (2D)
Single Vertex Translated with
Random Falloff Selected
Circle of Influence

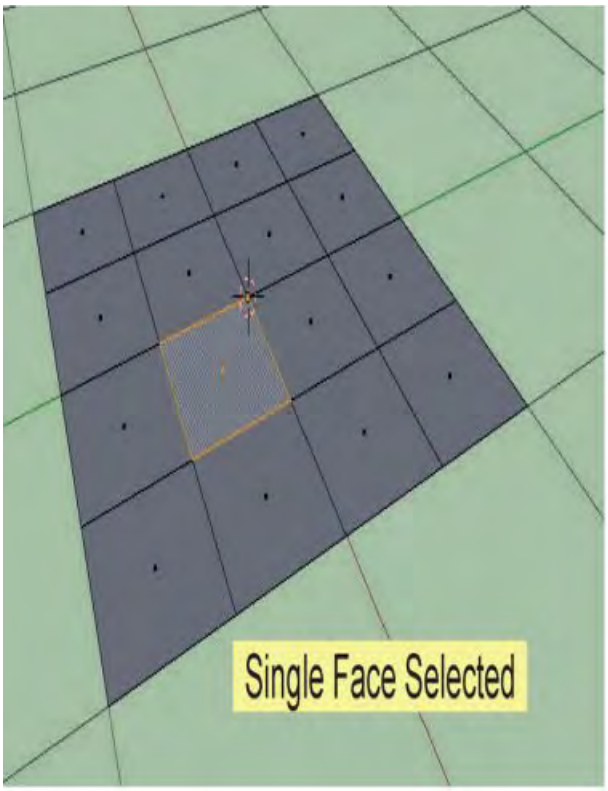
Figure 5.40

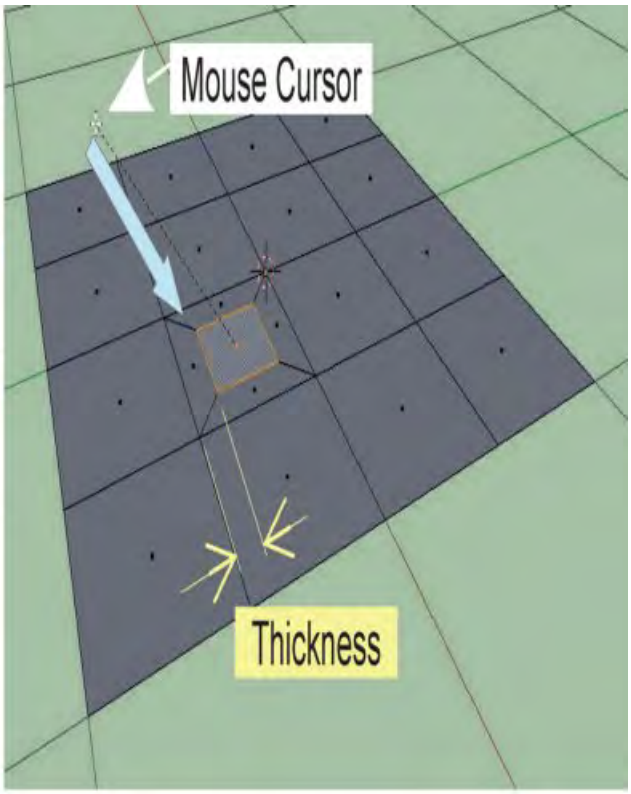
5.11 Inset Faces

The **Inset Faces** command causes new Faces to be created inside or outside a selected geometry.

To demonstrate, delete the default Cube in the 3D Viewport and add a **Plane** Object. Zoom in (Scroll MMB), Tab into **Edit** mode and Subdivide the Plane (Number of Cuts = 3). Deselect the Vertices (LMB Click in Editor). Change from **Vertex Select** mode to **Face Select** mode.

Select a single Face and with the Mouse Cursor in the 3D Viewport Editor, **moved to one side**, press the **I Key** and move the Mouse Cursor towards the center of the Face.(Figure 5.41). Alternatively, with a **Face** selected, click **Face** in the 3D Viewport Editor Header and select **Inset Faces** Move the Mouse towards the center of the Face. Another method is, click **Inset Faces** in the **Tool Panel**. Click on the yellow paddle that displays and move it towards the center. With either method, and you see new Faces created. Click LMB to set the new Faces in position.


Single Face Selected


Mouse Cursor
Thickness

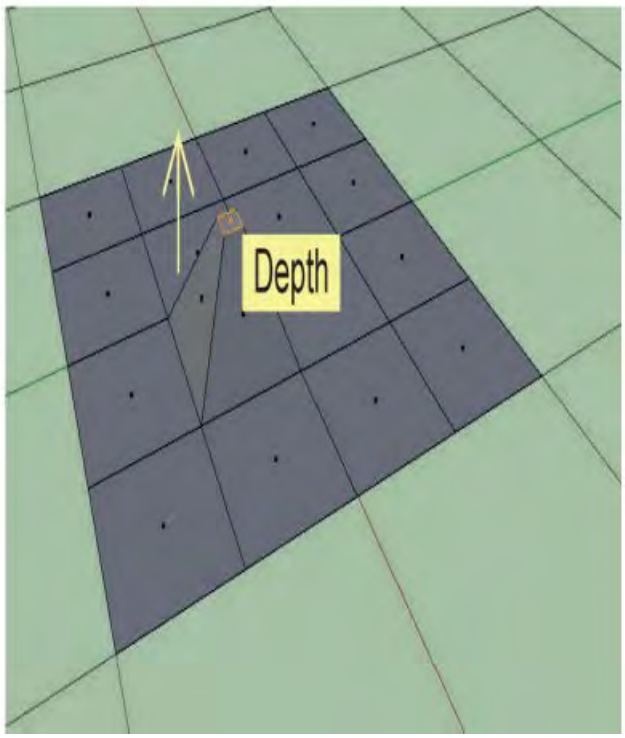
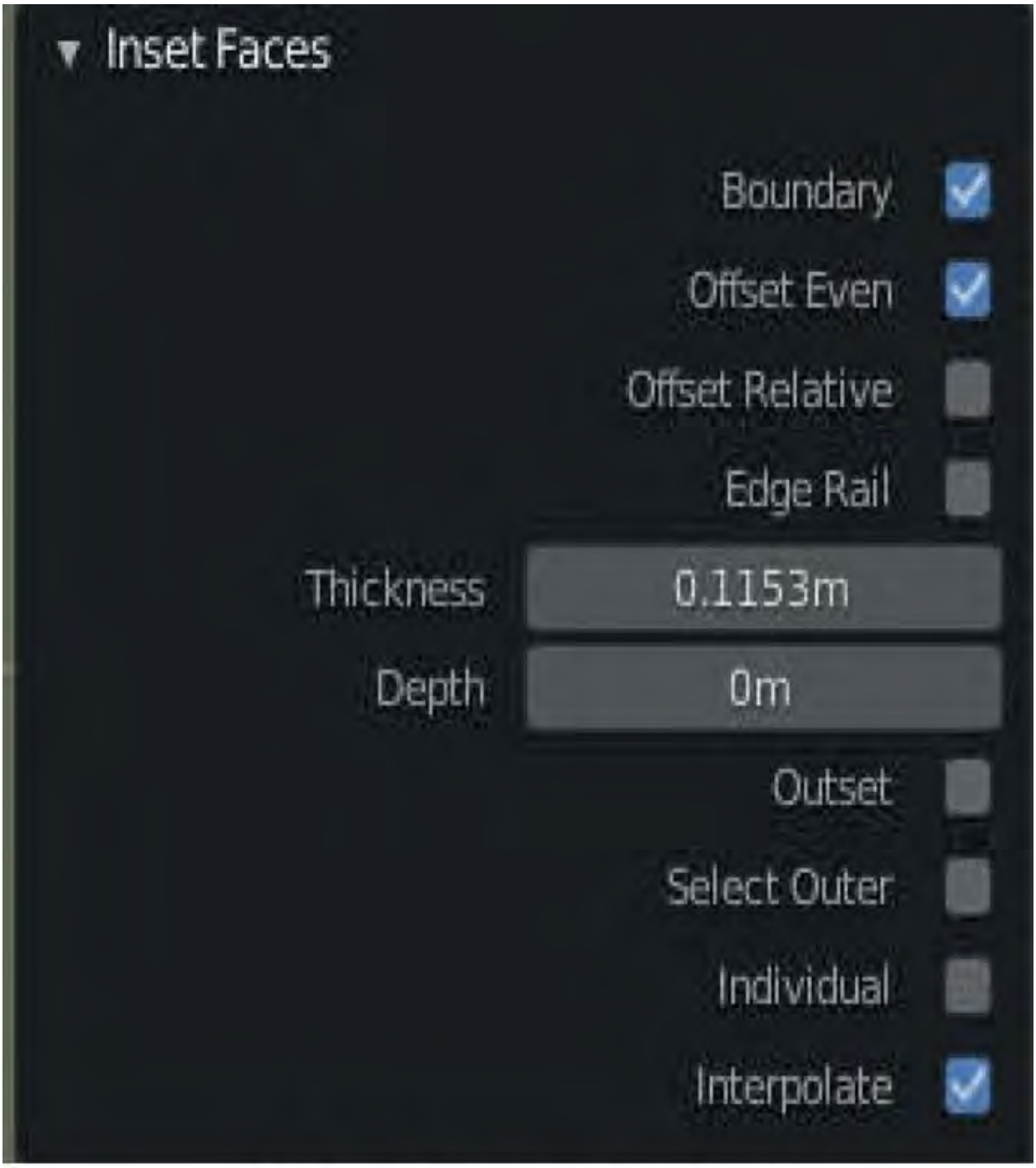

Depth

Figure 5.41

Note: Inset Faces displays in the **Last Operator Panel** (Figure 5.42). While this panel is displayed you can make adjustments. The **Thickness Slider** controls the size of the new Face. The **Depth Slider** displaces the face normal to the Plane. Positive values above the Plane, negative values below the Plane.



Last Operator Panel

Figure 5.42

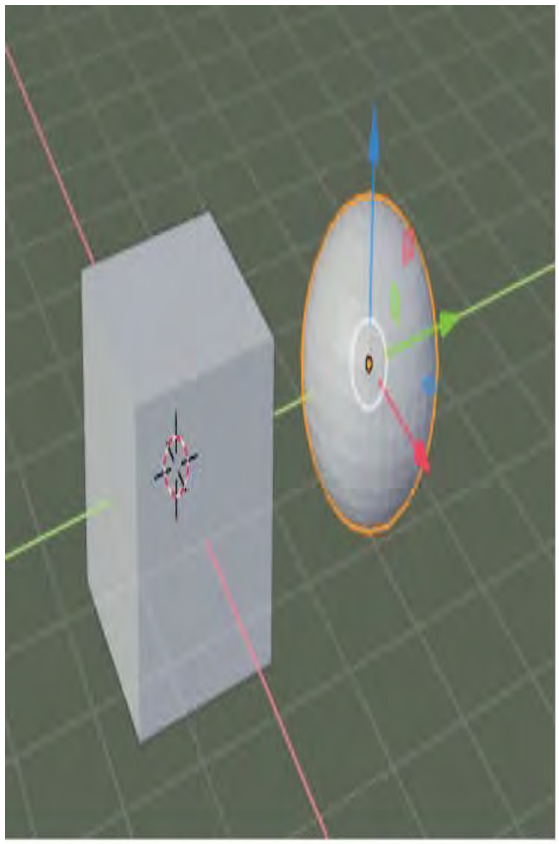
Note: The procedure can be processed in both **Vertex Select Mode** and **Edge Select Mode** by selecting the perimeter of an area on the surface of the plane. The principle can be applied to the surface of any mesh Object.

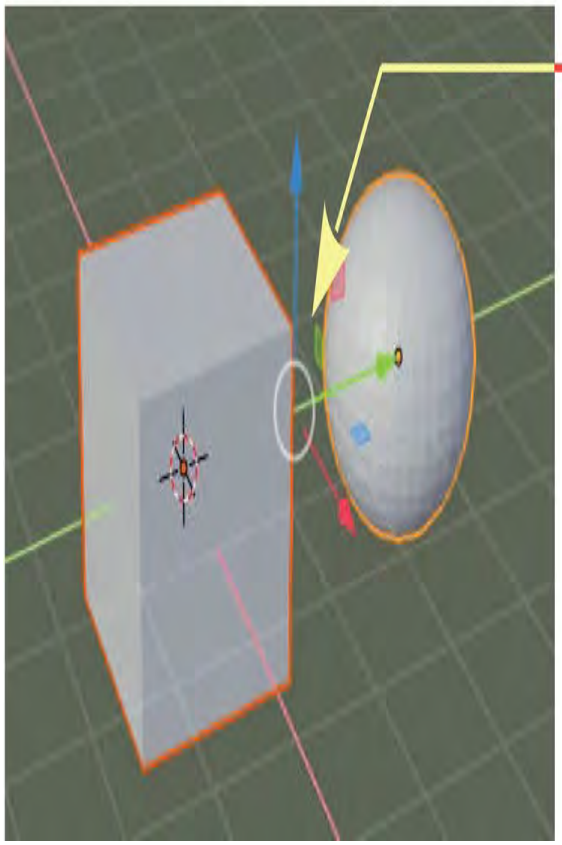
5.12 Parenting

In **5.8 Joining and Separating**, it was demonstrated how two Mesh Objects may be joined together to form a single combined Object. There are, however, occasions when you will want two or more Objects to act as a single unit but not actually have the mesh joined. For example, you may want one Object to follow another when the first Object is moved. This can be achieved by **Parenting** or creating a **Child Parent Relationship** between Objects.

To demonstrate, add a UV Sphere Object to the default Scene (Figure 5.43).

Note: In the diagram the Move Tool in the Tool Panel has been activated and the Manipulation Widget employed to position the UV Sphere.





With the Move Tool activated displaying the Manipulation Widget, you will observe that the **Widget relocates mid way** between the two Objects. This indicates that Parenting is in place. Using the Widget to move along the Y Axis will demonstrate that the Objects are Parented. They both move together.

Figure 5.43

With the UV Sphere selected press Shift and select the Cube. For the demonstration make the selection in this order (Sphere selected – Shift select the Cube).

Press **Ctrl + P Key** and click on **Set Parent To Object.**

> **Note:** The selection order in the Parenting operation is important.

Having the UV Sphere selected then selecting the Cube has made the UV Sphere the child of the Cube. Selecting the Cube individually and moving, will see the UV Sphere follow. If the UV Sphere is selected individually it can be moved independently.

6 Editing Tools

6.1 The Tool Panel
6.2 The Add Menu
6.3 The Last Operator Panel
6.4 Extrusion
6.5 The Extrude Region Tool
6.6 Inset Faces
6.7 The Inset Faces Tool
6.8 The Bevel Tool
6.9 Edge and Loop Selection
6.10 The Loop Cut Tool
6.11 The Knife Tool
6.12 The Poly Build Tool
6.13 The Spin Tool
6.14 Creating a Spin Profile
6.15 Spin Duplication
6.16 The Screw Tool
6.17 The Smooth Tool
6.18 The Edge Slide/Vertex Tool
6.19 The Shrink Fatten Tool
6.20 The Shear Tool
6.21 The Rip Region Tool

Editing Tools

In the previous chapter some of Blender's basic mesh editing techniques were introduced which allowed you to manipulate and reshape Primitives. These techniques are the beginning of the process and are essential for mastering more sophisticated practises. Blender incorporates many techniques for performing a variety of functions some of which are automated. The automated techniques may be considered as **Tools**.

Like any trades person, mechanic or engineer, knowing what tools are available, how each functions and most importantly what tool to use for a particular application is the key to success. And, yes! where to find the tool.

As you have seen many editing techniques are simple Key and Mouse commands applied to an Object or to an Object's mesh surface. Other tools automate some fantastic and complex operations. In this chapter some of the Tools will be introduced with examples showing how they are employed.

Other Tools will be introduced in context with specific operations as you progress through the book.

6.1 The Edit Mode Tool Panel

The **Tool Panel** at the left hand side of the **3D Viewport Editor** in **Edit Mode** contains a of tools for automating editing processes (Figure 6.1).

The top eight Tools are identical to those found in the Object Mode Tool panel and function in the same way when applied to a selection of Vertices, Edges or Faces. The Tools are activated by clicking (LMB) on a Tool (highlights blue – see **Select Box Tool** at the top of the Tool Stack).

Figure 6.1

To gain Screen space you may mouse over at the side of the Tool Panel to reveal a **double headed arrow**, click, hold and drag, to expand or reduce the size and arrangement of the panel (Figure 6.2).

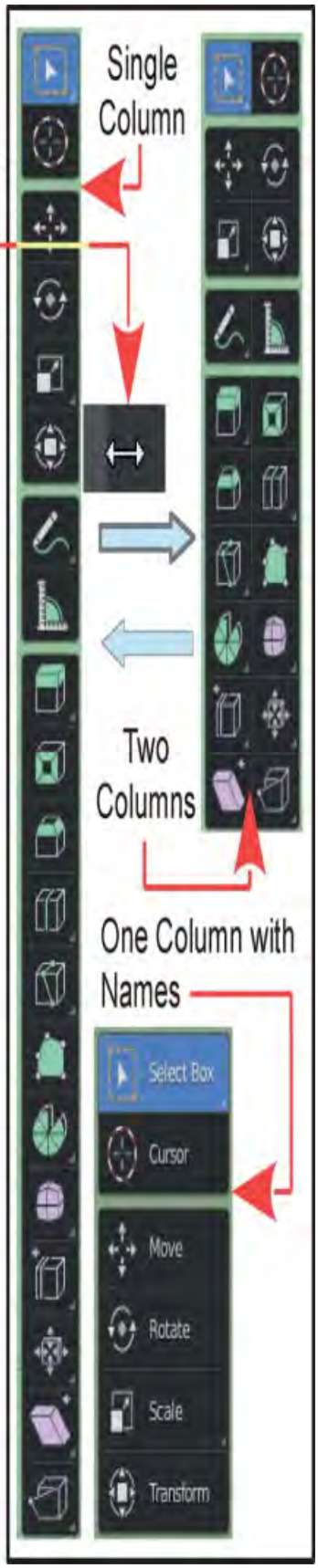


Figure 6.2

By default the Tool Panel displays with a single Column as shown in Figure 6.2 (LHS). Mouse over on the panel edge and drag the double headed arrow to expand. First expansion created a two column display. Second expansion displays names as shown in Figure 6.1.

Note: You cannot apply the Rotate, Scale or Ruler Tools to a single Vertex.

Pressing the **T Key** on the Keyboard toggles hide and show the Tool Panel.

Note: When in Object Mode you can not enter Edit Mode unless you have a Mesh Object in the 3D Viewport Editor. With two or more Objects in the 3D Viewport Editor, the selected Object or the last Object that was selected will be the active Object when entering Edit Mode.

6.2 The Add Menu

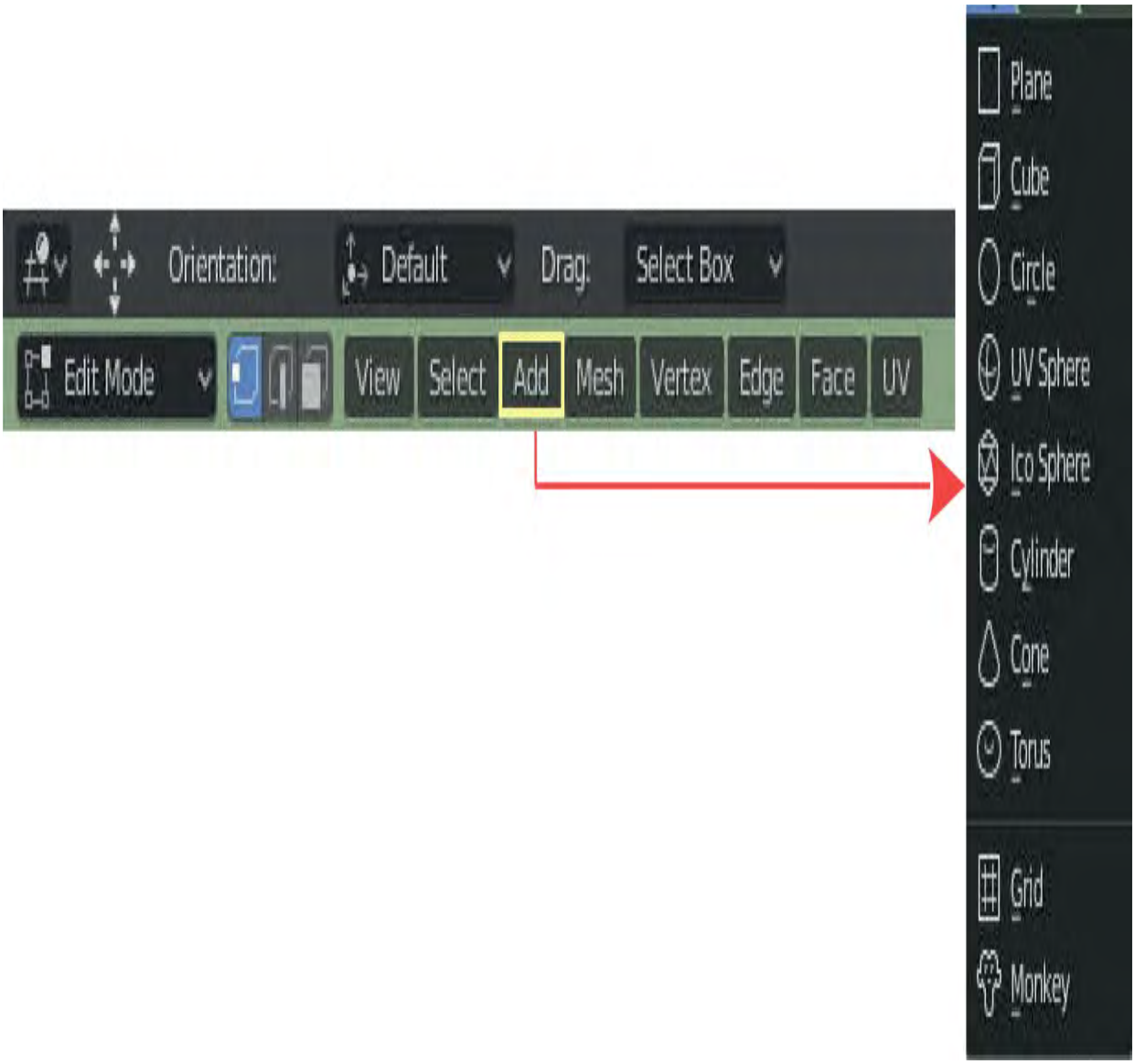


Figure 6.3

Objects may be added to the Scene with the **3D Viewport Editor in Edit Mode** by clicking **Add** in the 3D Viewport Editor Header (Edit Mode).

Remember: Objects added to a Scene in **Edit Mode** are automatically joined to the last object selected while in **Object Mode** (see Chapter 5 – 5.8).

6.3 Last Operator Panels

When an action has been performed in the 3D Viewport Editor, the **Last Operator Panel** is displayed in the bottom LH corner of the Editor. **Click to Expand**, allowing values in the panel to be adjusted affecting the action (see Chapter 4 – 4.13).

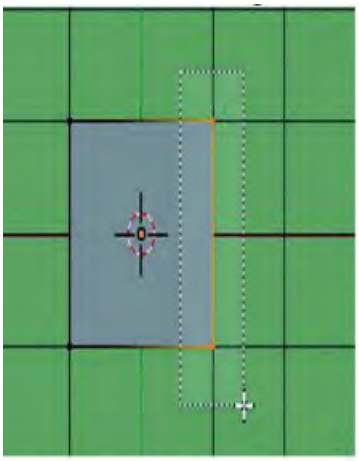
The Last Operator Panel (Figure 6.4) shows that a Cube Object was added at the center of the Scene (Location X, Y, Z = 0 m)

Figure 6.4

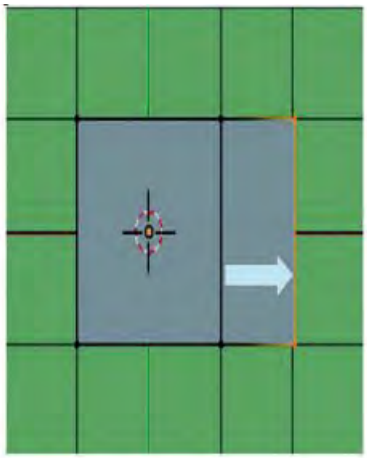
6.4 Extrusion

Before examining the **Extrude Region Tool** consider Extrusion in general.

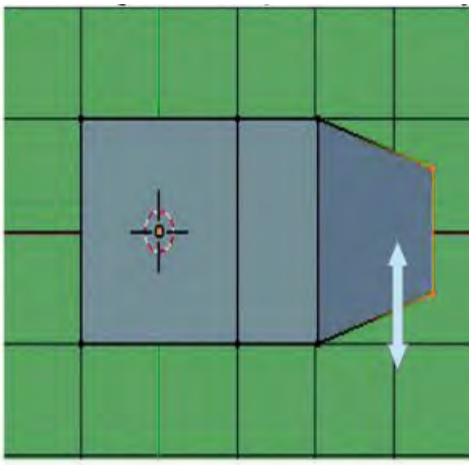
Extrusion means, stretching such that the shape is altered. Shapes can be altered by selecting a single Vertex or a group of Vertices, then Translating, Rotating or Scaling. You may also select Edges or Faces and apply the same processes (Figure 6.5).



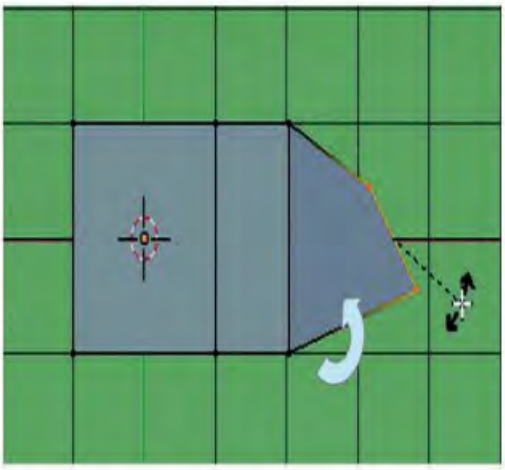
Vertices Selected



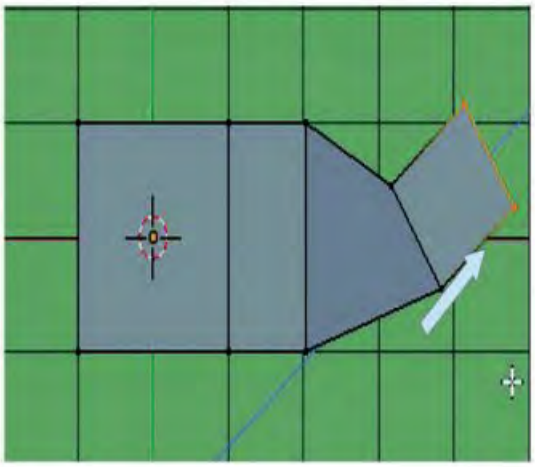
Vertices Extruded



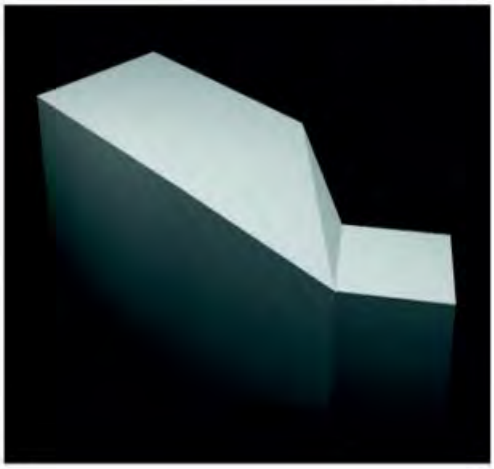
Extruded and Scaled



Vertices Rotated



Vertices Extruded



Rendered Image of Camera View

Figure 6.5

Vertices are selected with the Object in Edit Mode. When the selected **Vertices** are **Extruded duplicate Vertices are created**, which may be Moved, Scaled or Rotated. The new Vertices remain attached to the original Object's surface mesh thus altering the shape of the Object.

For Extrusion instruction see the following page.

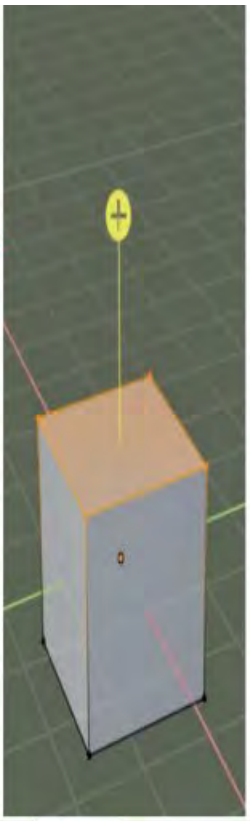
In Figure 6.5 the RH Vertices of a Cube (Top Orthographic View) are selected in **Edit Mode** by pressing the **B Key** (Box Select) and dragging a rectangle (don't forget to have **Toggle X-Ray on** - Chapter 5 – 5.2 – Figure 5.3). With the Vertices selected press the **E key** (Extrude) + **X Key** (confines Extrusion to the X Axis) and drag the Mouse. The Vertices are duplicated and repositioned. With the Vertices remaining selected the Extrusion process is repeated, and the selection Scaled down by pressing the **S Key** and moving the Mouse in. The selection is Rotated by pressing the **R Key** and moving the mouse. Finally, the selection is Extruded again.

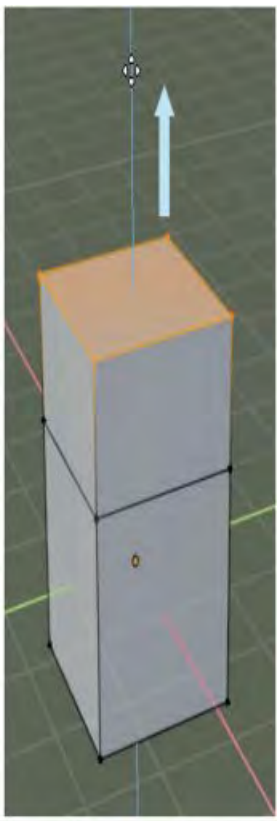
6.5 The Extrude Region Tool

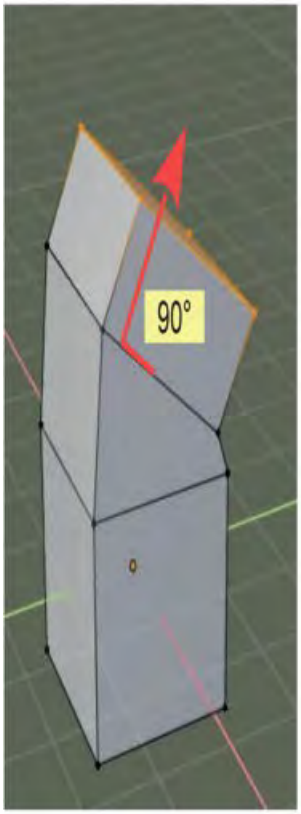
Activating the **Extrude Region Tool** displays a manipulation Widget located at the center of the selection. In 6.6 the selection is the top Face of the default Cube. Click, hold and drag the Widget to Extrude the selection.

The Tool has four options. Mouse over on the Tool and click and hold to expand the panel.

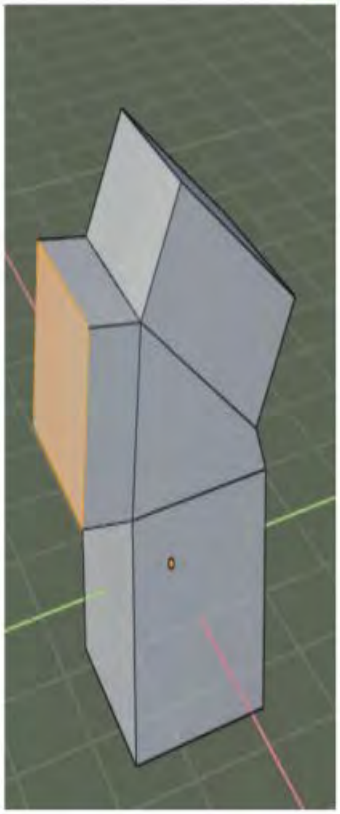
Extrude Region



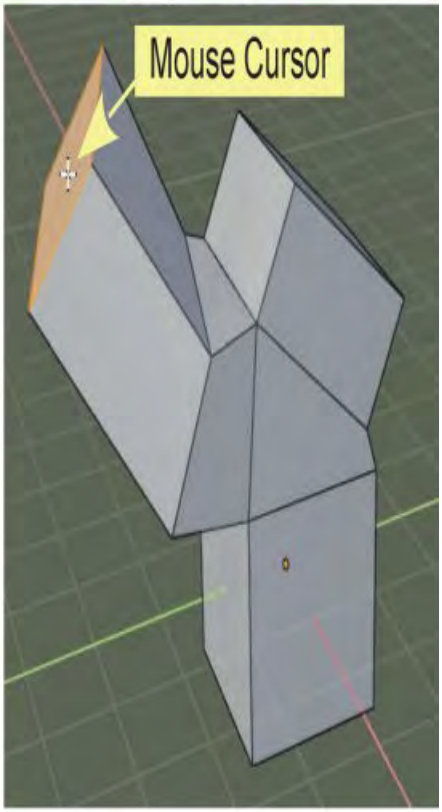




Extrude Along Normal



Extrude Individual



Extrude to Cursor

Figure 6.6

Extrude Along Normal means Extrude at right angles to the selected Face. **Extrude Individual** allows Extrusion of an individual Face.

With **Extrude to Cursor** selected the selected Face will be Extruded to wherever you click the Mouse Cursor (click the Select Box Tool at the top of the Tool Panel to cancel).

Note: The Extrusion methods can be applied to Faces, Edges and Vertices.

6.6 Inset Faces

Inset Faces creates new Faces inside a selected geometry (see 5 – 5.11).

To demonstrate, select the top **Face** of the default Cube while in Edit Mode. Press the **I Key** (**I**nsert - Figure 6.7) and move the **Mouse Cursor** towards the center of the selected Face. You see new Faces created. Click LMB to set the new Faces in position.

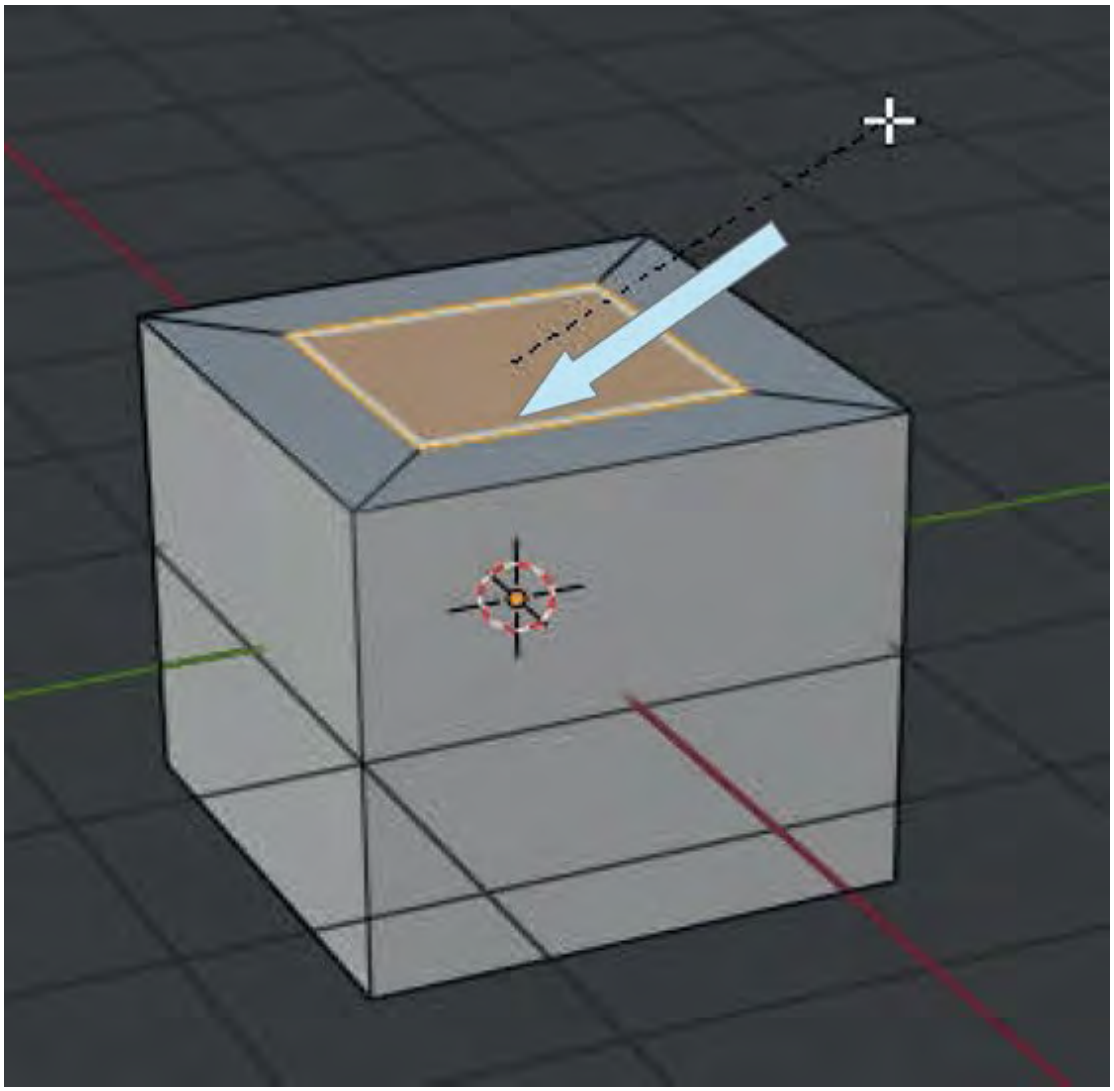


Figure 6.7

Note: The **Inset Faces Last Operator Panel** displays (Figure 6.8). While this panel is displayed you can make adjustments. The **Thickness slider** controls the size of the new Face. The **Depth slider** displaces the Face normal to the surface (positive values above the surface, negative values below). Note: The procedure can be processed in both **Vertex Select Mode** and **Edge Select Mode** by selecting the perimeter of an area on the surface. The procedure can be applied to the surface of any mesh Object.

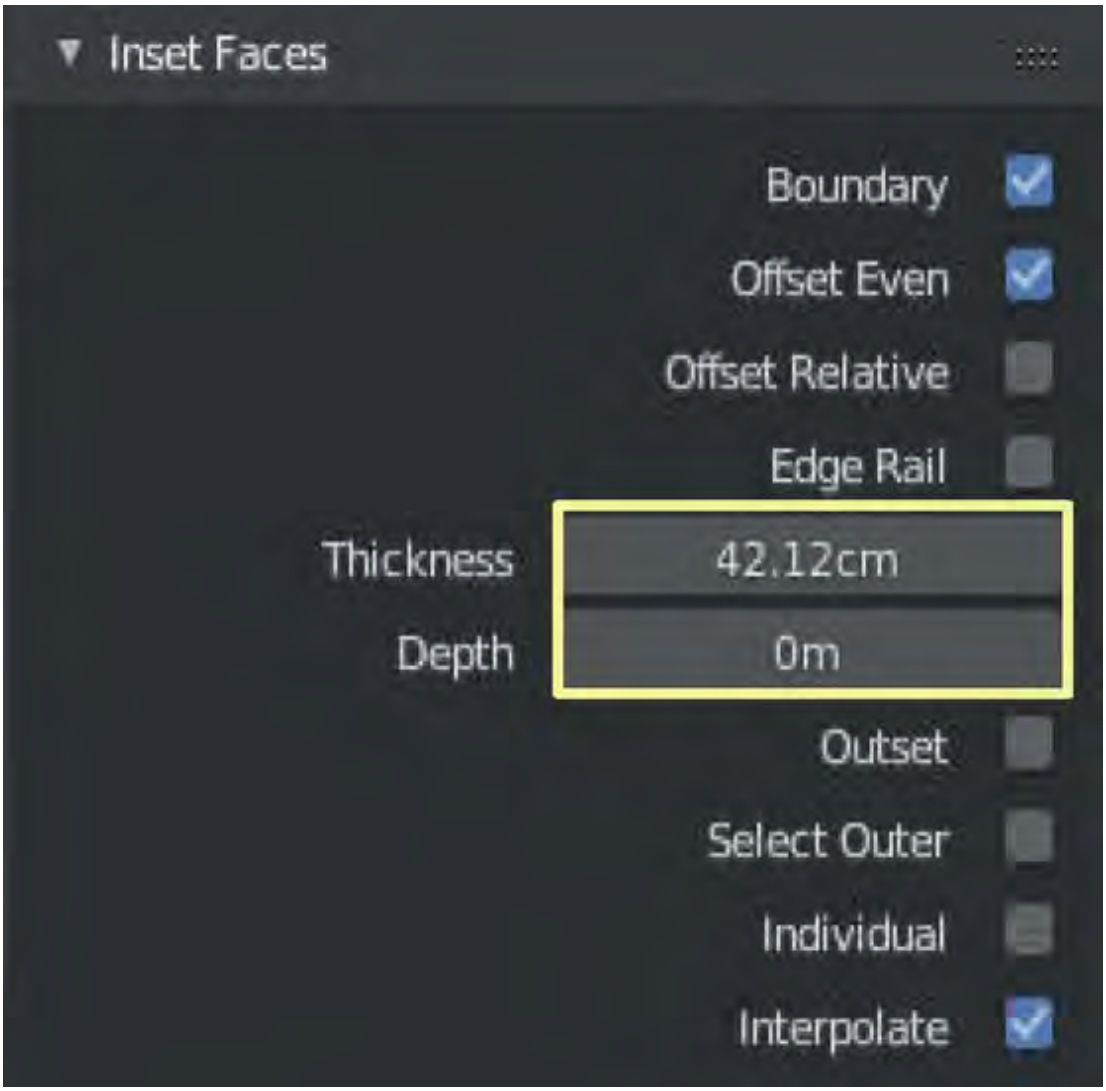
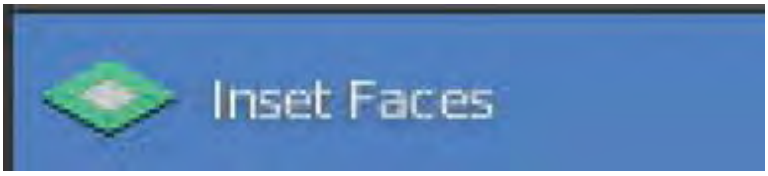


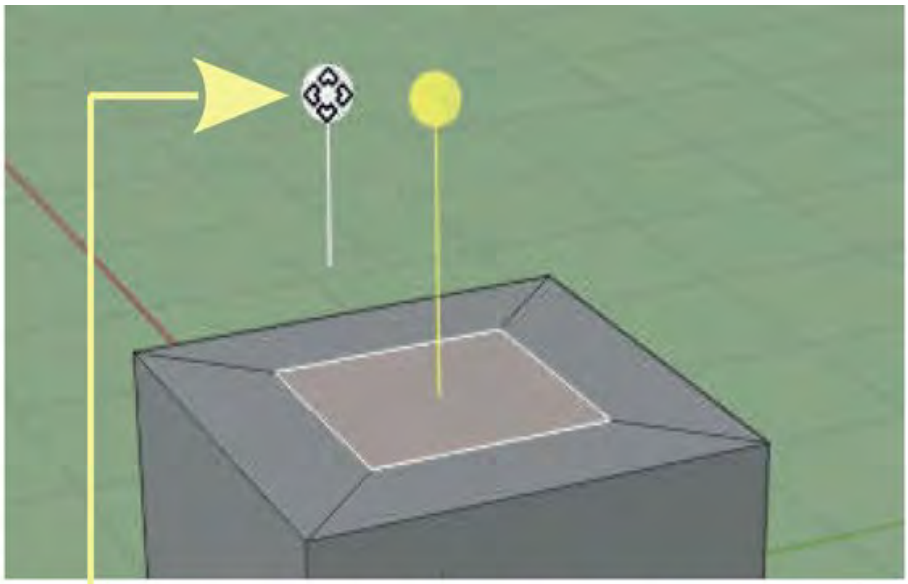
Figure 6.8

6.7 The Inset Faces Tool



Activating the **Inset Faces Tool** replicates the above procedure. Click the Tool to activate. Click hold and drag the yellow paddle that displays (Figure

6.9). While the Tool is activated you can select any Face in the 3D Viewport and apply the procedure.



Mouse Cursor Over the Paddle

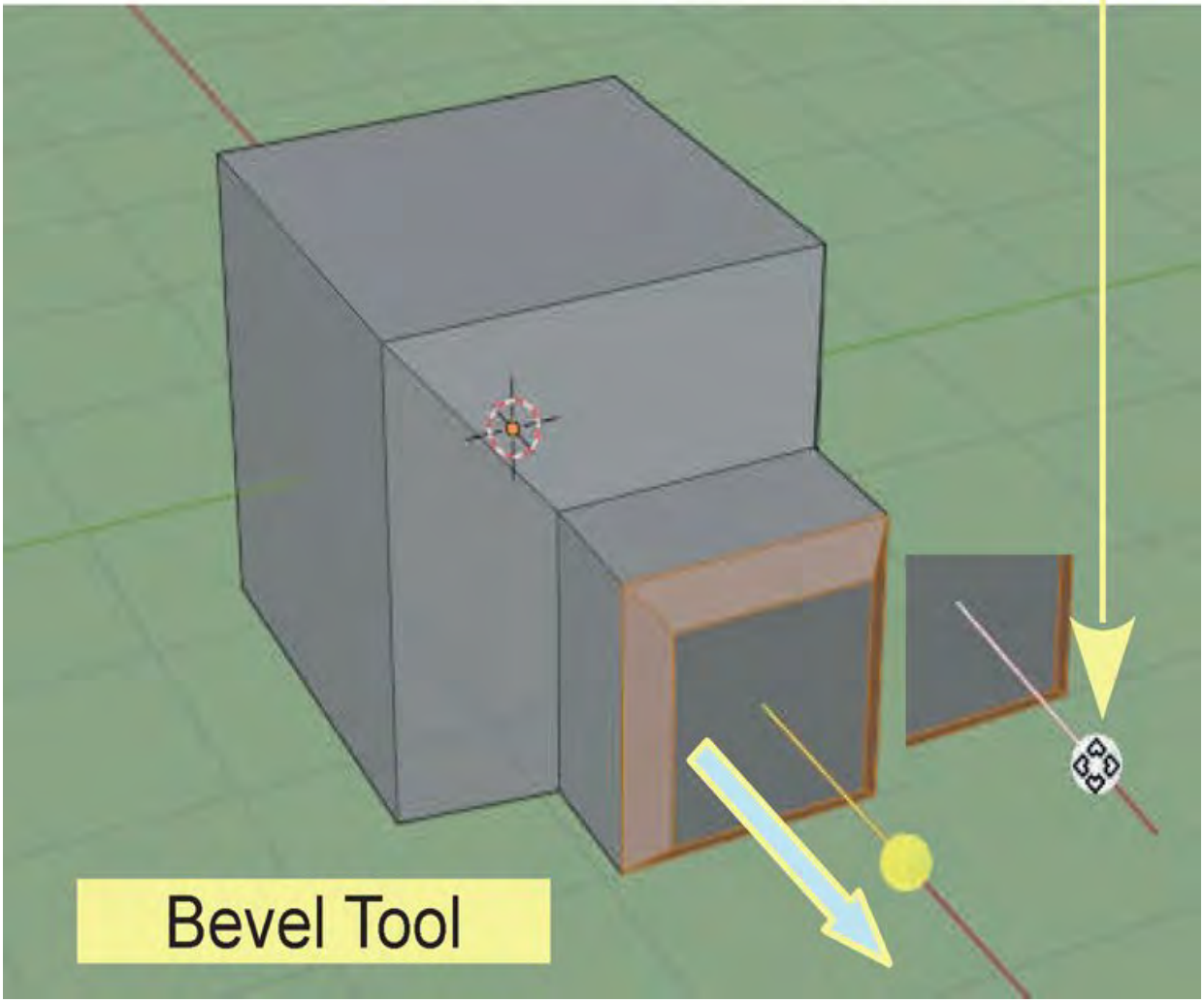
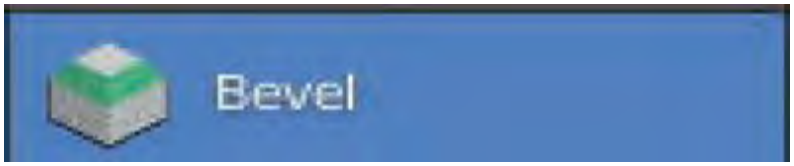


Figure 6.9

6.8 The Bevel Tool

The **Bevel Tool** bevels the Edges of a selected Face (Figure 6.10).



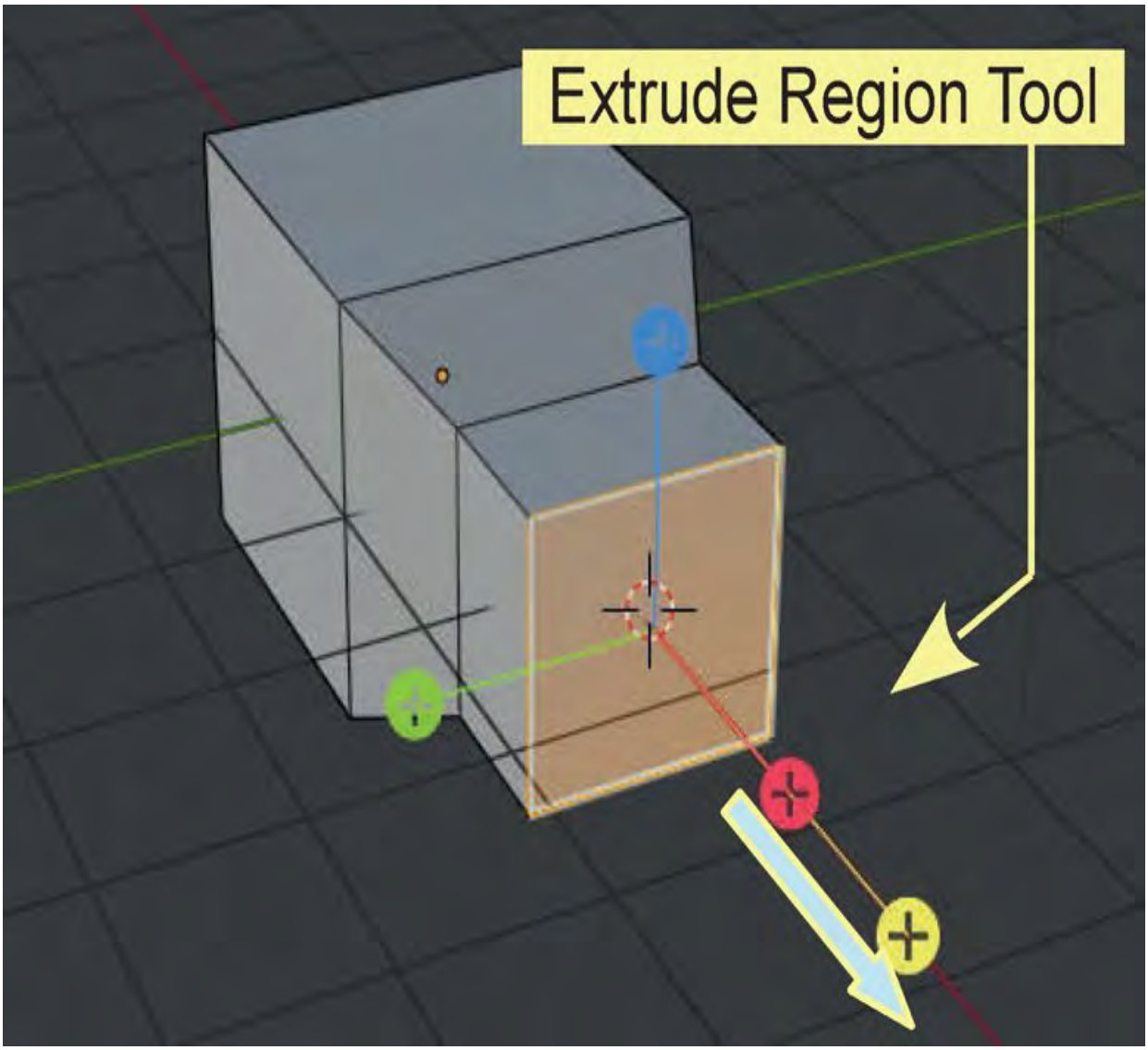


Figure 6.10

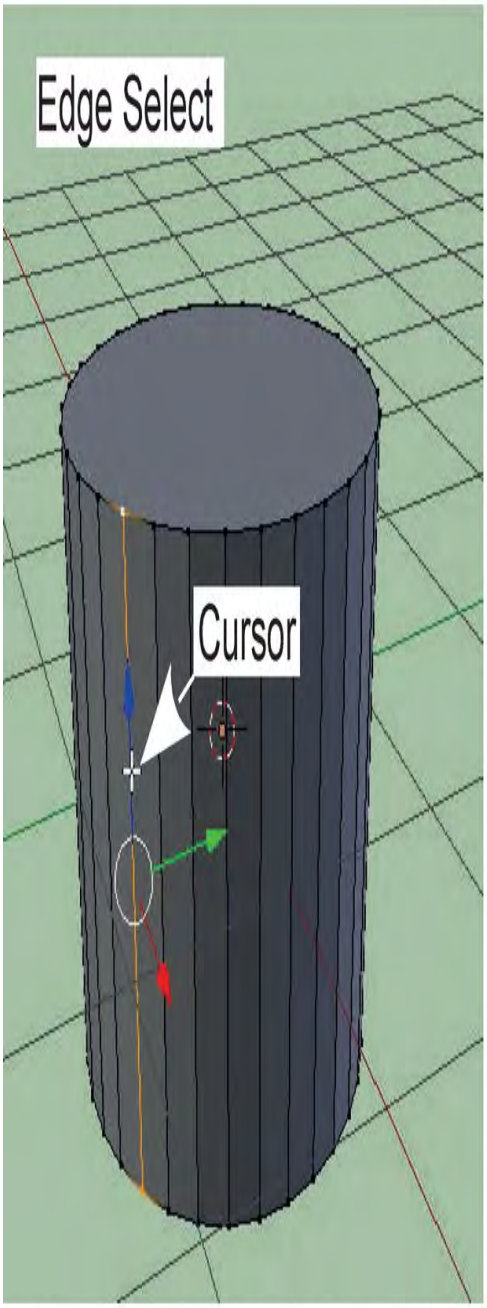
The default Cube with the front face Extruded and Scaled down then Extruded a second time.

With the face selected activate the Bevel Tool. Place the Mouse Cursor, click hold and drag to bevel the edges of the Face.

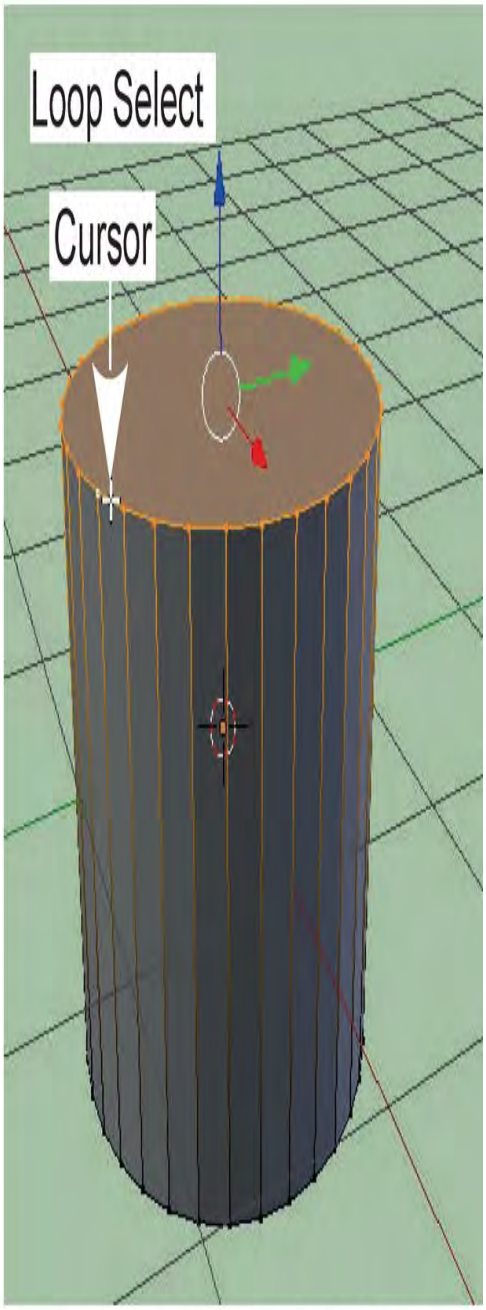
6.9 Edge and Loop Selection

When working with Vertices, it is sometimes useful to select Vertices which form an Edge or a

Cylinder Object in Edit Mode



Place the Mouse Cursor (White Cross) over an Edge – Press **Alt – Click LMB**



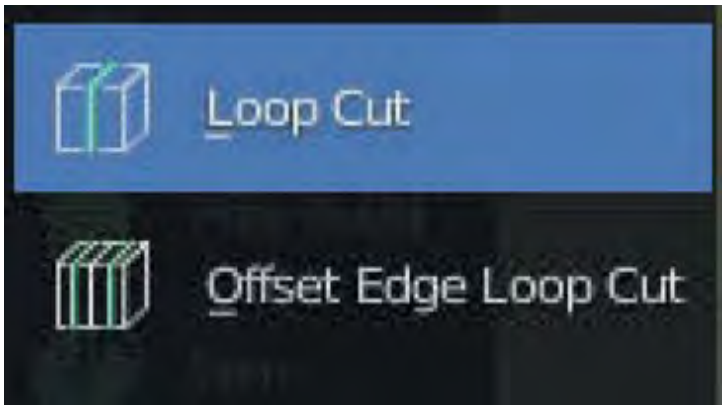
Place the Mouse Cursor (White Cross) over a Loop – Press **Alt – Click LMB**

Figure 6.11

6.10 Loop Cut Tool

The **Loop Cut Tool** allows you to create new Edges and Loops without selecting Vertices.

Note: This Tool has two options;



Mouse over on the Loop Cut Tool in the Tool Panel, Click LMB and hold to see the options. Drag the Mouse Cursor over the option required and release to select the option.

Have the Object selected in Object Mode, Tab to Edit Mode and deselect all Vertices. To demonstrate, a Cylinder will be used, in Vertex Select Mode.

With the Loop Cut option selected, you may create a new **Edge** or a new **Edge Loop**.

To Create a New Edge (Vertices connected by a vertical Edge), hover the Mouse Cursor (white cross) over either the upper or lower ring of Vertices to

display the new intended Edge as a vertical yellow line between any two Vertices on either ring. You position the yellow line between sets of Vertices line by dragging the Mouse (Figure 6.12a). Click LMB to set the new Edge. By default the Edge is positioned mid way between Vertices. If you wish to locate the new Edge other than mid way, Click LMB, hold and drag the mouse (Figure 6.12b, c), release to set. You can only drag the new Edge in between two vertices.

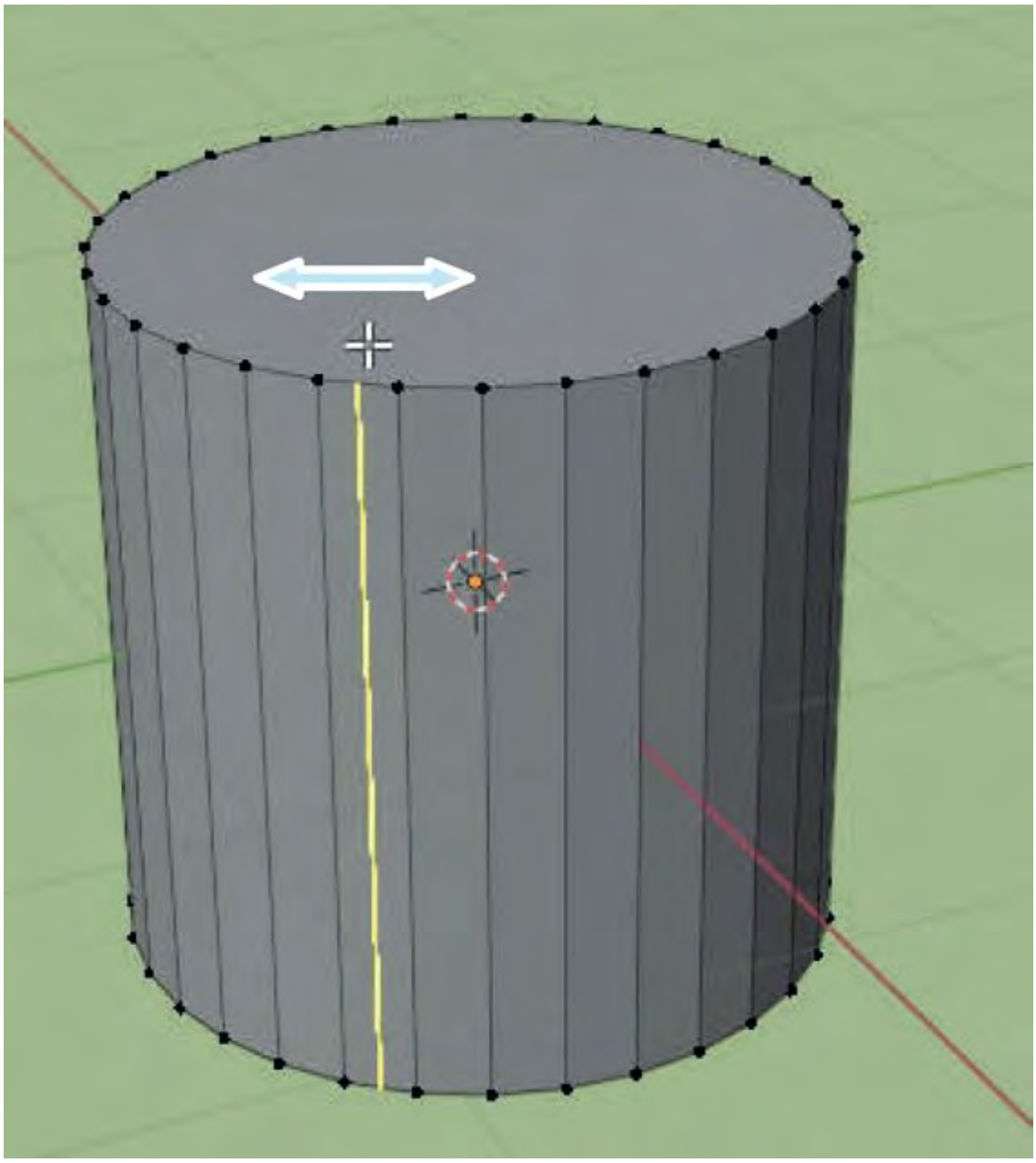


Figure 6.12a

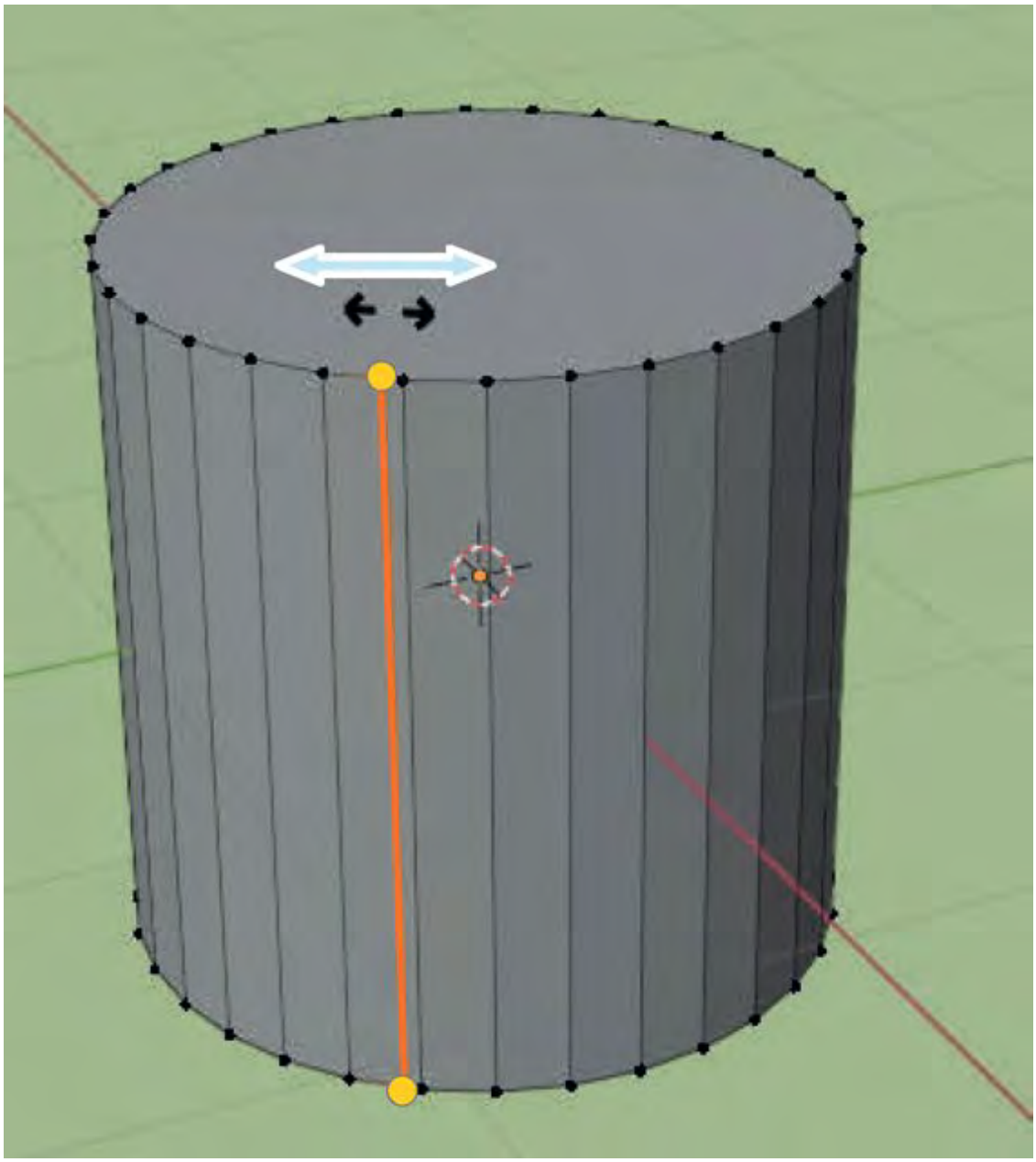


Figure 6.12b

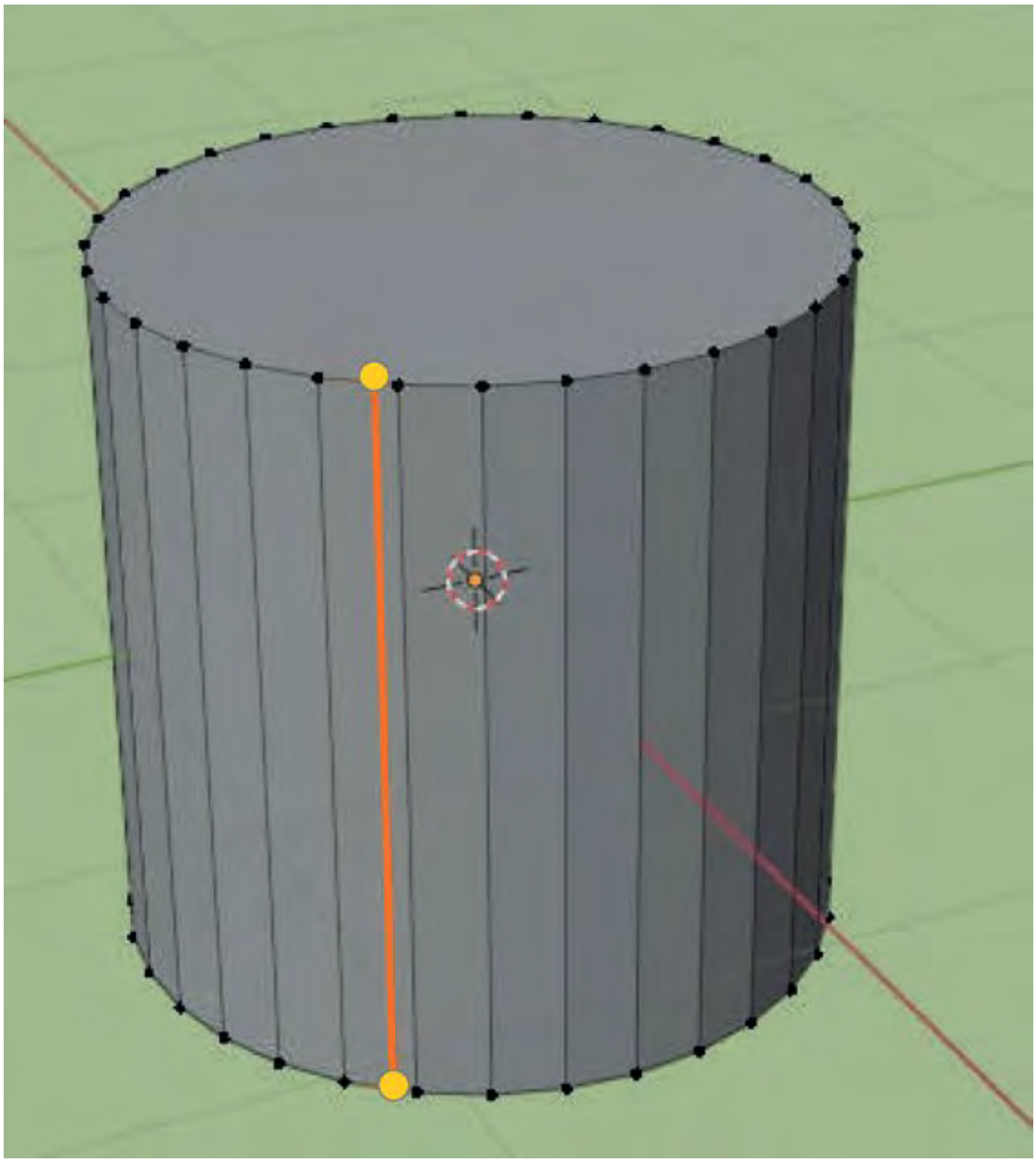


Figure 6.12c

To Create a New Edge Loop mouse over (white cross) on the side of the Cylinder to display the new Edge Loop as a yellow circle (Figure 6.13a). By default the yellow circle will display mid way between the upper and lower Edge Loops or mid way between any loops previously created. Click LMB, hold and drag the Mouse to position the new Edge Loop and release to set in position (Figure 6.13b, c).

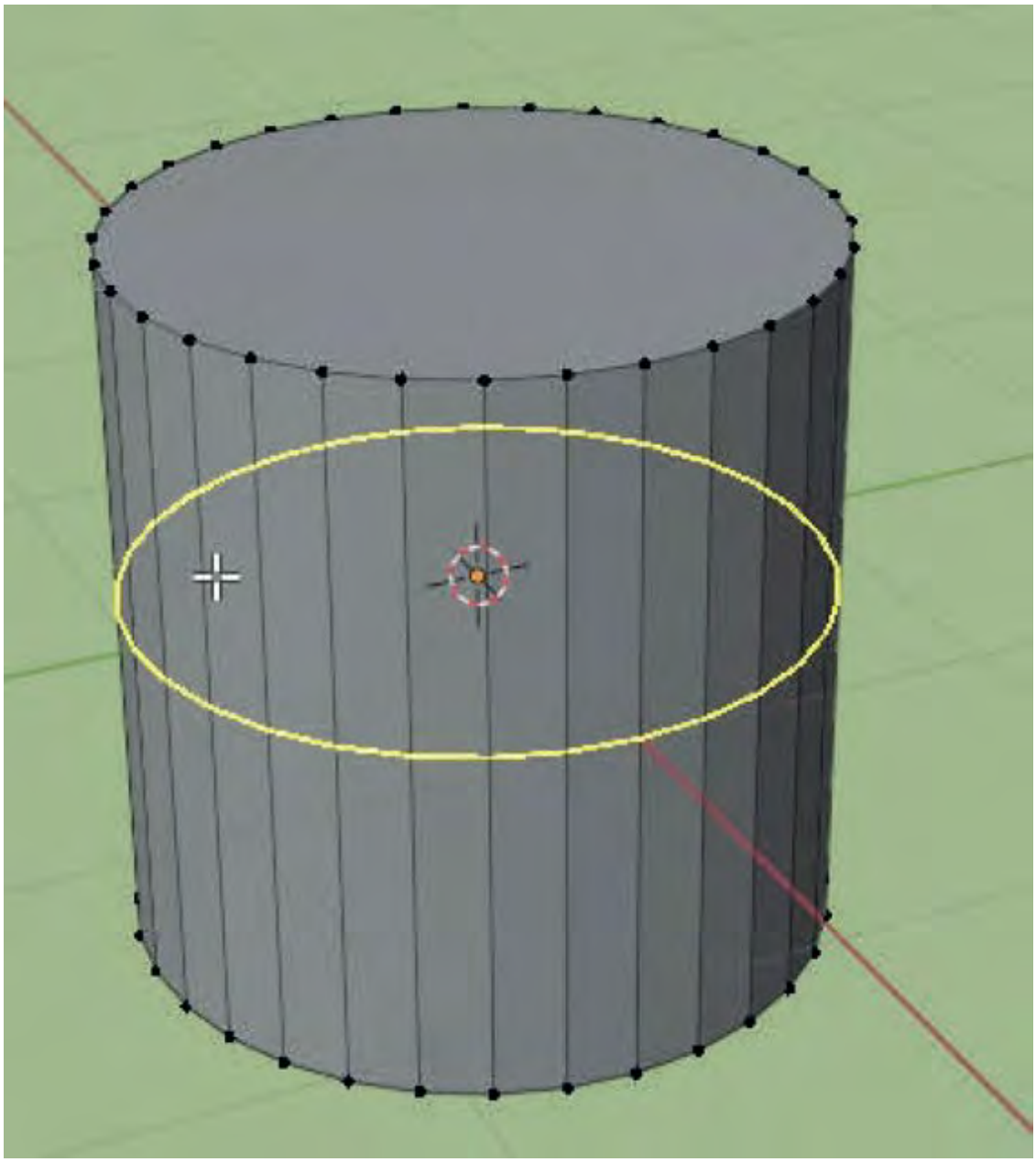


Figure 6.13a

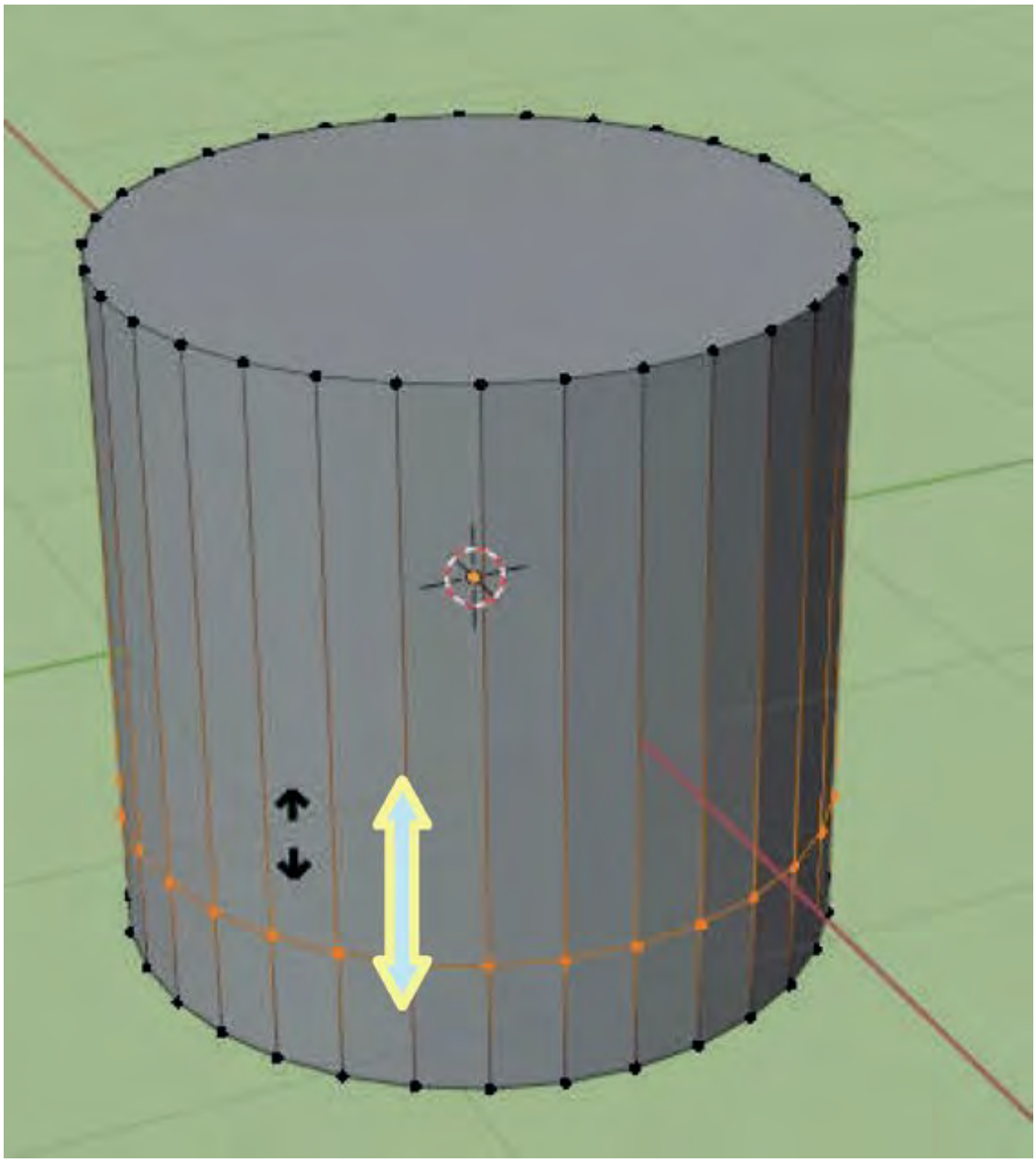


Figure 6.13b

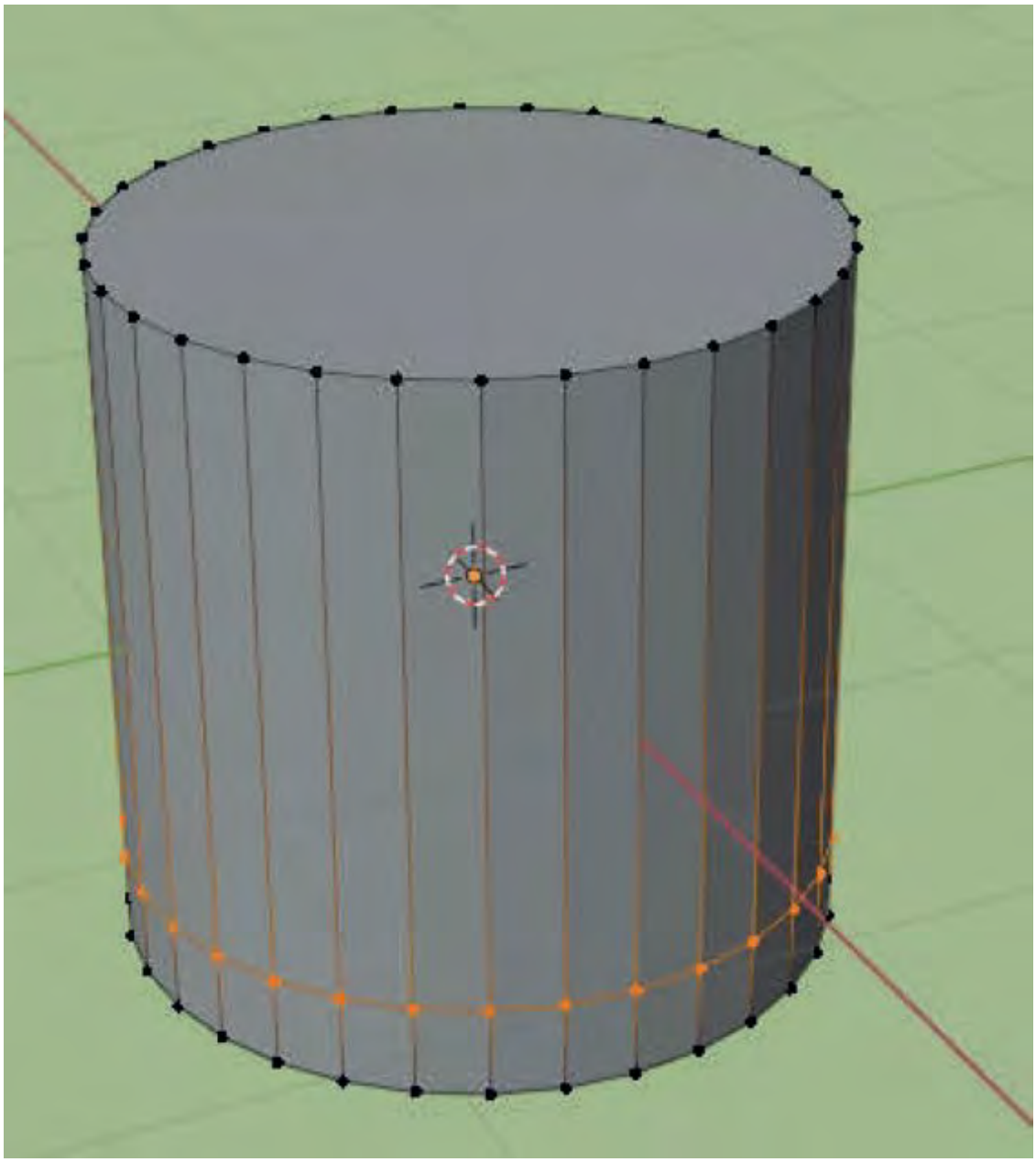


Figure 6.13c

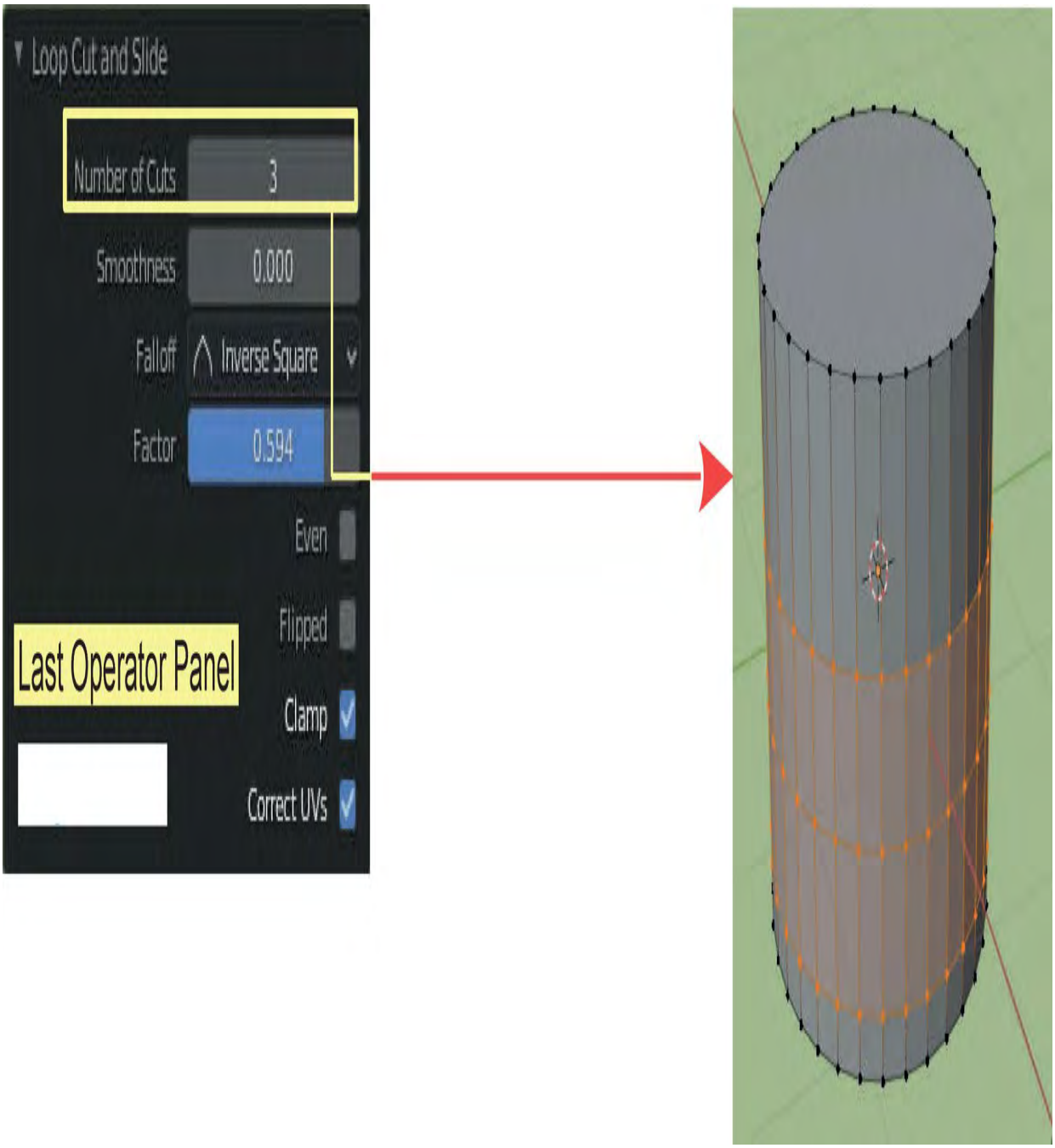


Figure 6.14

After setting the new Loop Cut in position you can create additional Loop Cuts by adjusting the Number of Cuts in the Last Operator Panel (Figure 6.14).

Alternatively you can preset the number of Loop Cuts to be created in the 3D Viewport Header (Figure 15).

Figure 6.15

Offset Edge Loop Cut

Having created a new Edge Loop you may duplicate Edge Loops either side of the cut using the Offset Edge Loop Cut Tool. Activate the Tool, position the Mouse Cursor adjacent to the cut, clickand hold LMB and drag the Mouse (Figure 6.16).

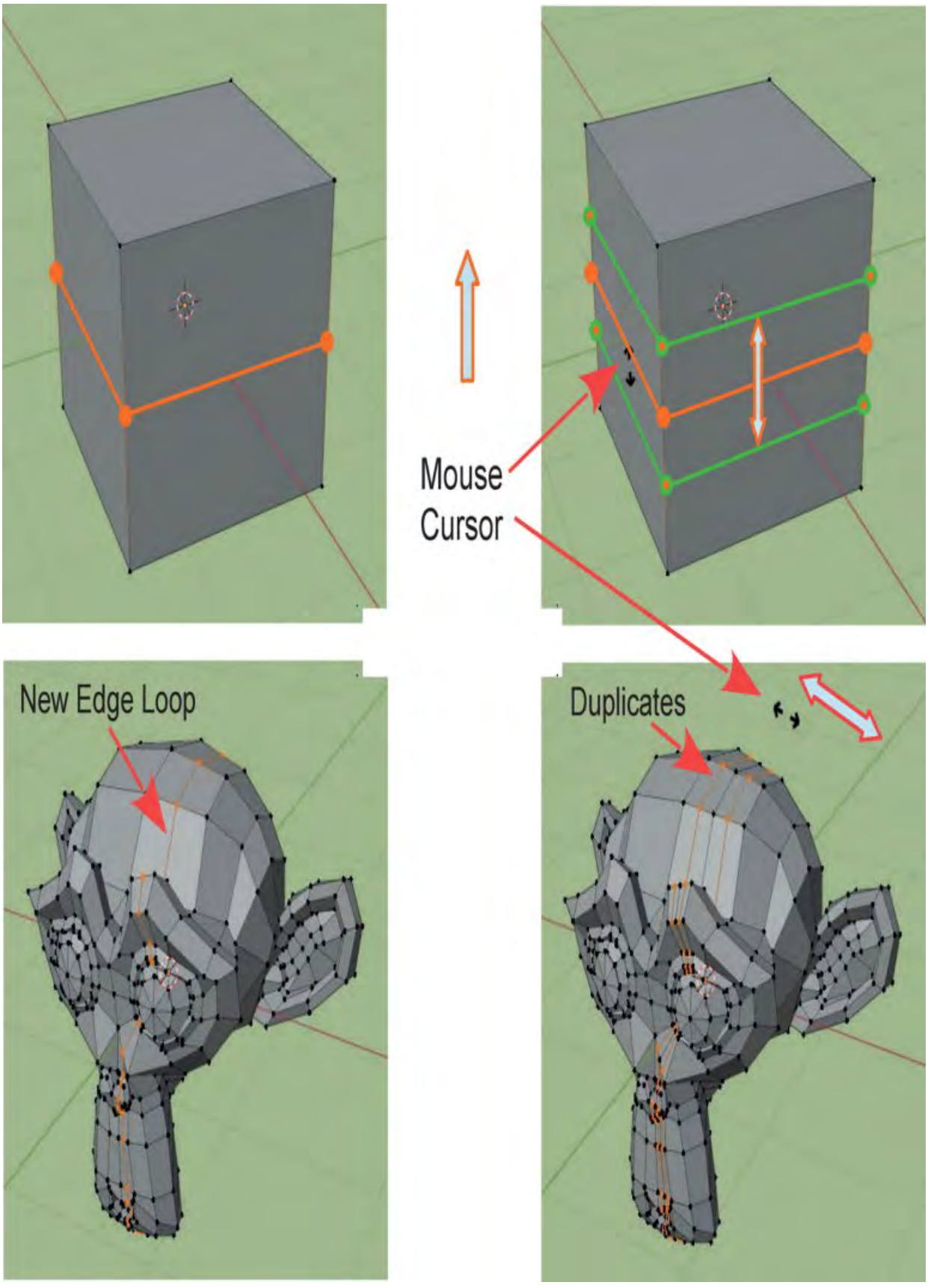

Mouse
Cursor
New Edge Loop
Duplicates

Figure 6.16

6.11 The Knife Tool

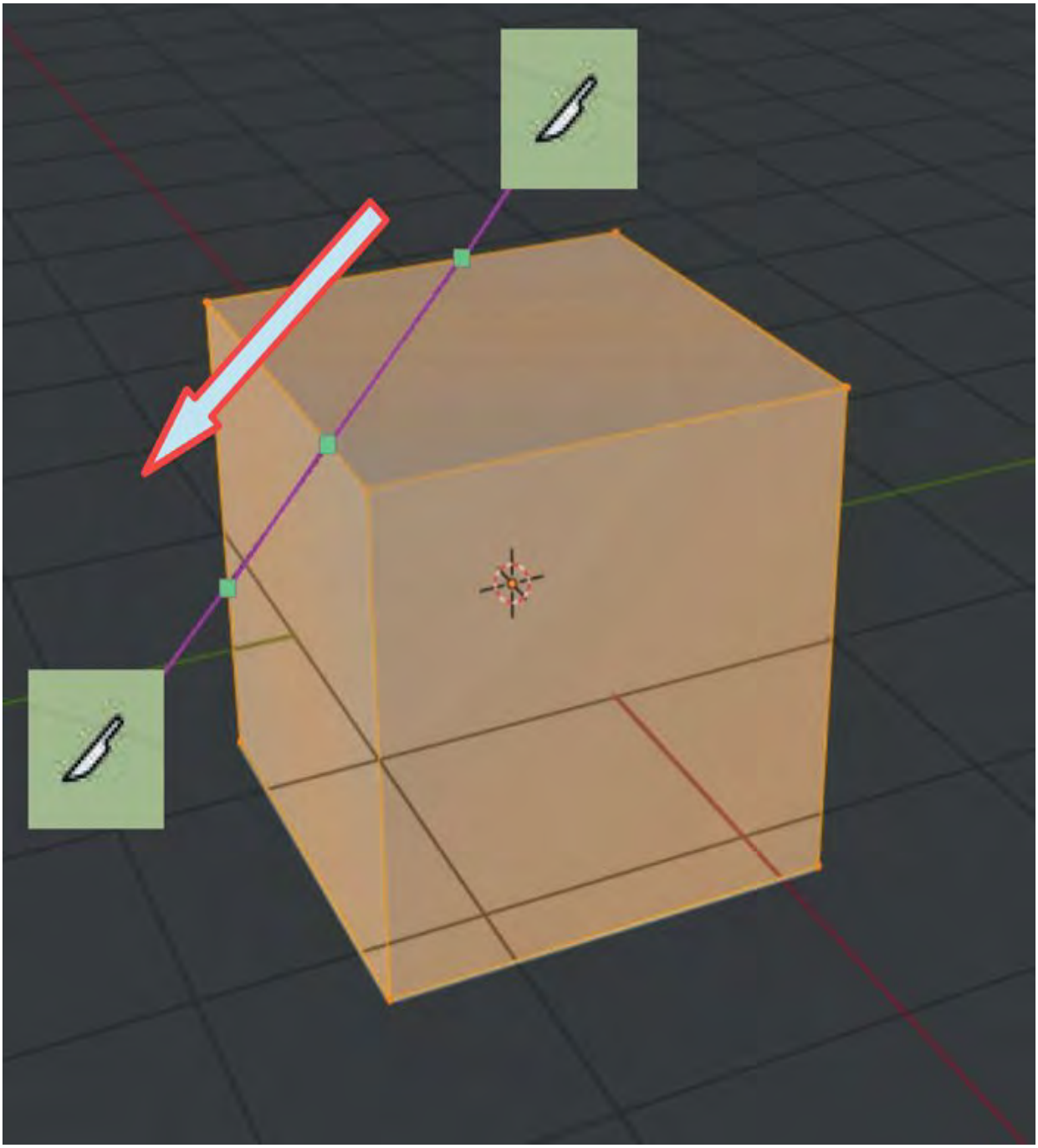
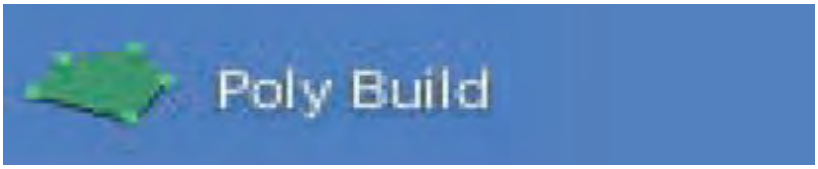


Figure 6.17

With the **Knife Tool** activated clicking LMB in the Editor displays the Mouse Cursor as a knife. Hold and drag across Edges creating Vertices and Edges. Release the Mouse button and press Enter.

6.12 The Poly Build Tool



Poly Build allows you to create Polygon Surfaces generating shapes by simply clicking LMB in the 3D View Editor.

With the **Poly Build Tool** activated, in **Vertex Select Mode**, Mouse Over on an **Edge** (highlights). Click **LMB, hold and drag** to generate a new Face. Click LMB to place.

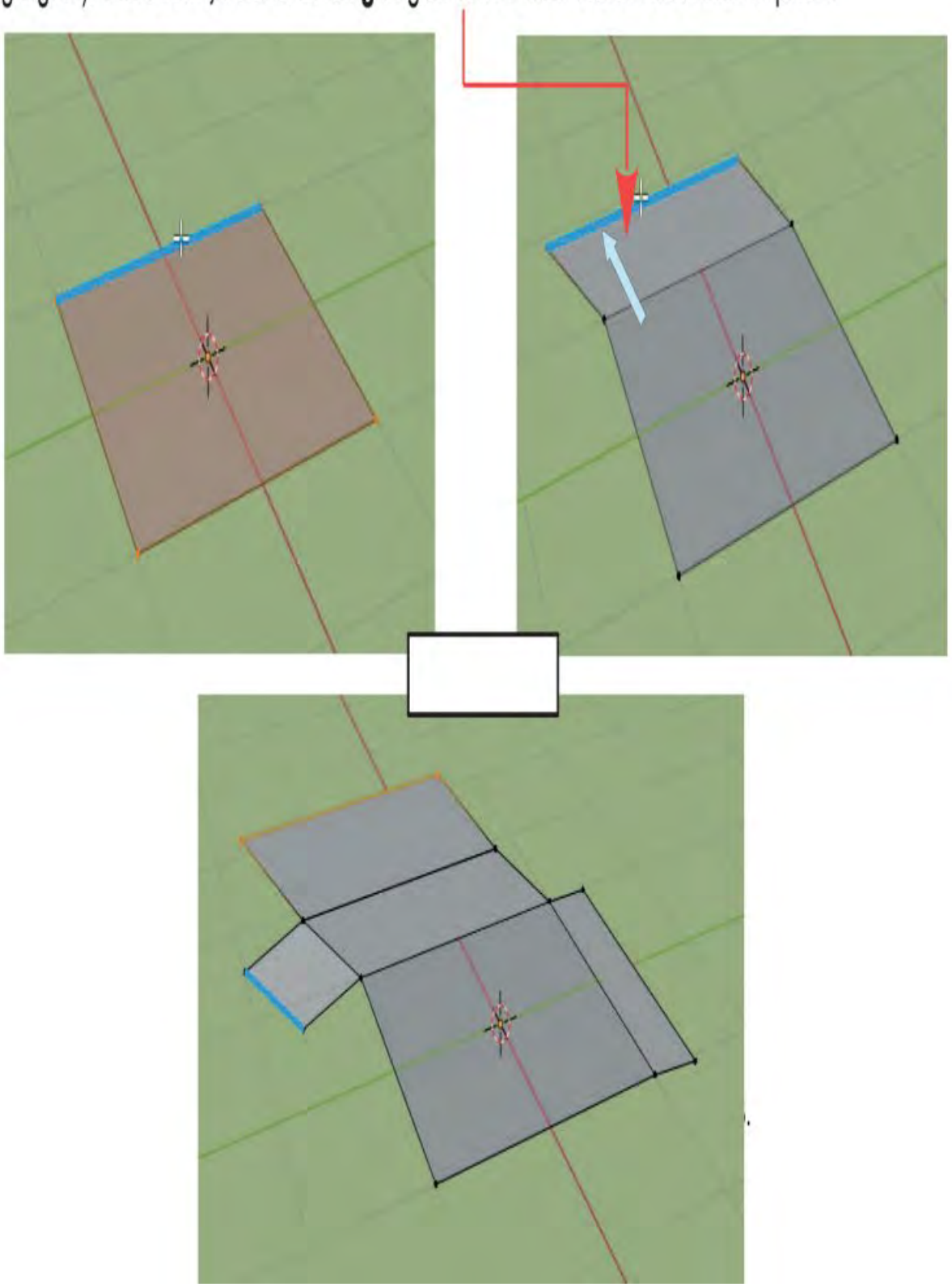
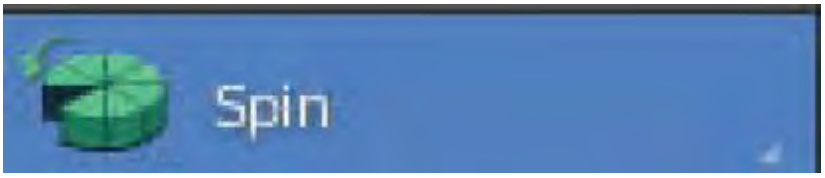


Figure 6.18

6.13 The Spin Tool



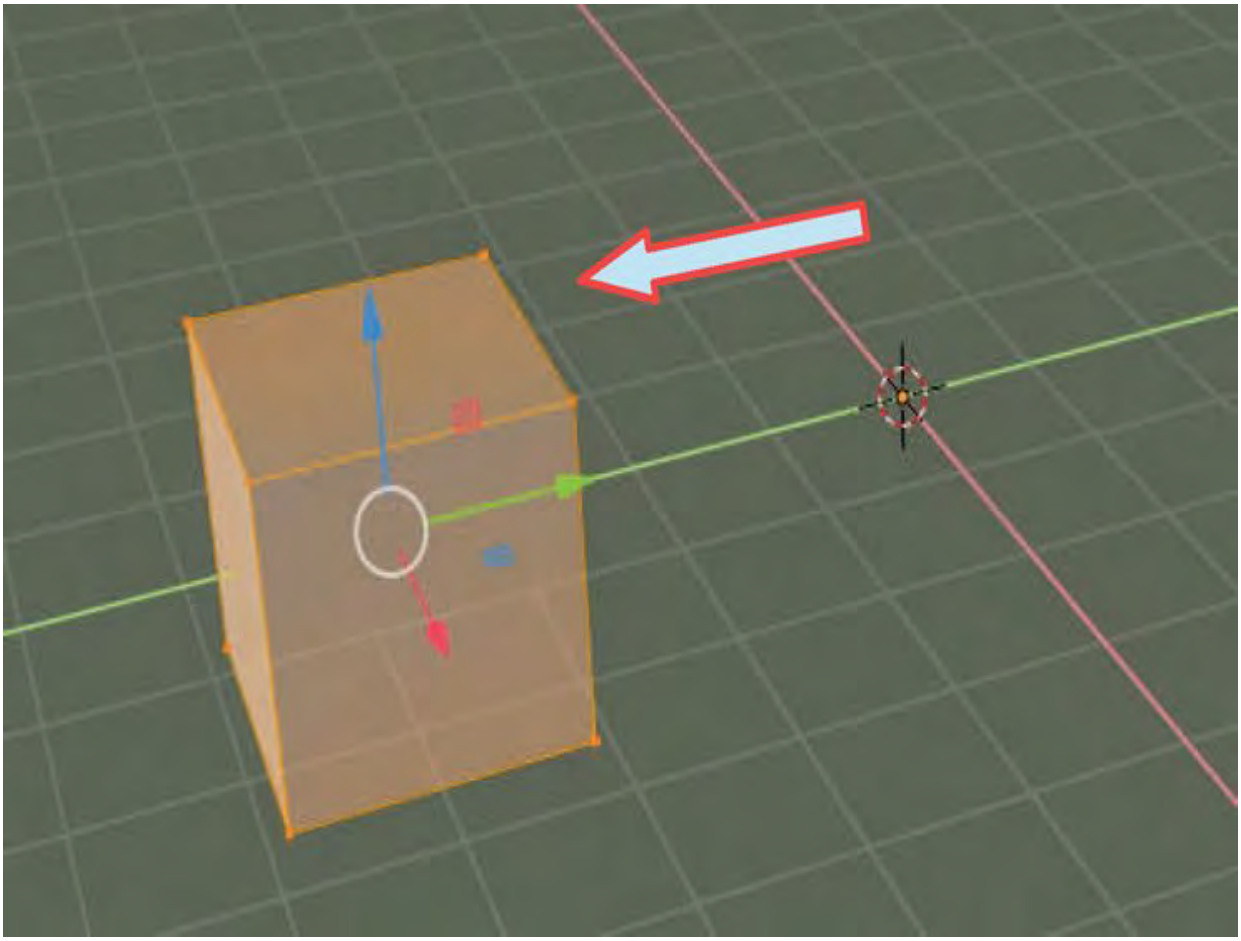


Figure 6.19

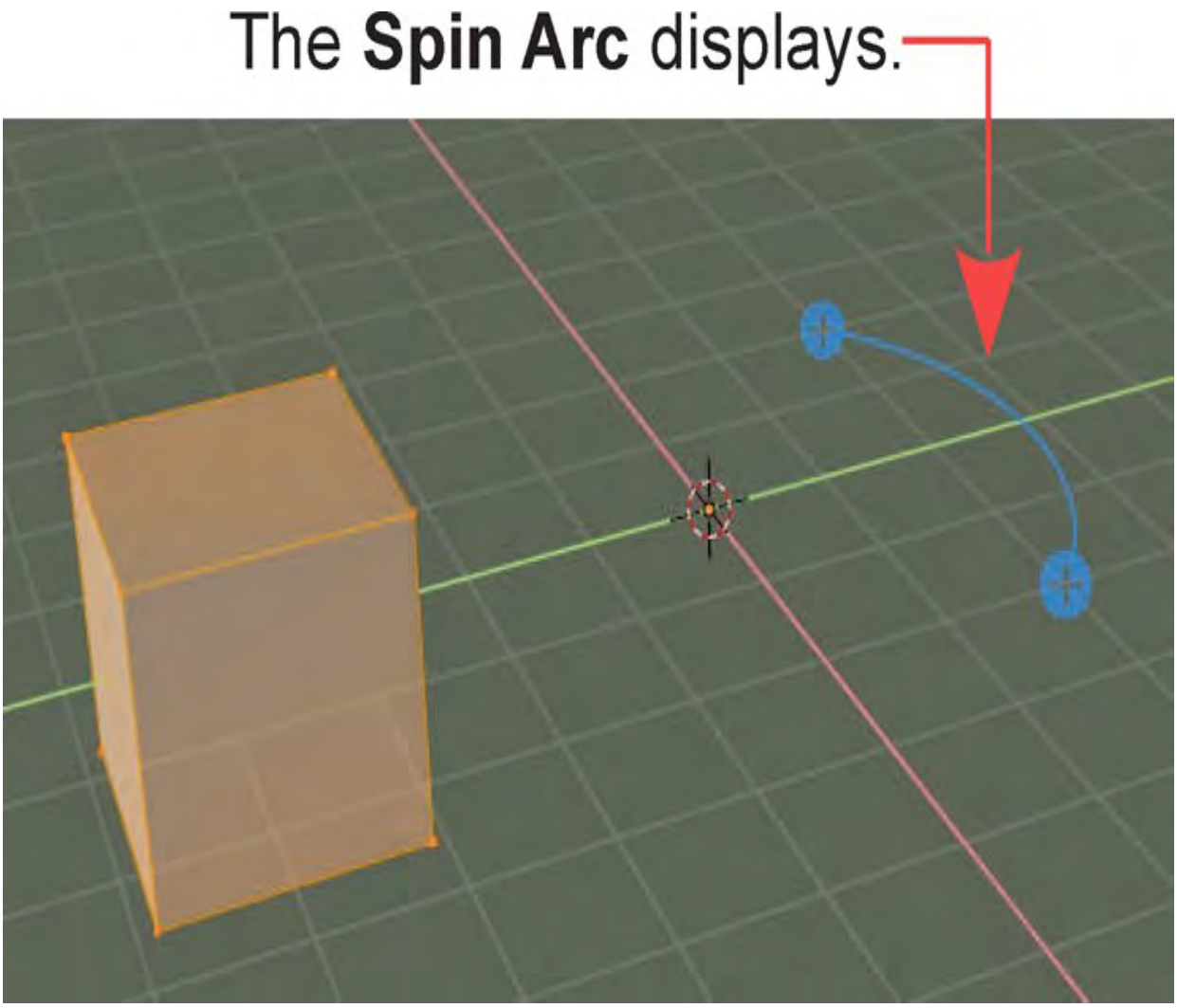


Figure 6.20

With all Vertices selected, Click LMB on a cross, hold and drag the Mouse to create a partial Spin (Figure 6.21. Release LMB. To Spin the Cube 360° about the center of rotation (the **3D View Editor Cursor**) (Figure 16.22) enter Angle 360° in the Last Operator Panel (Figure 16.23).

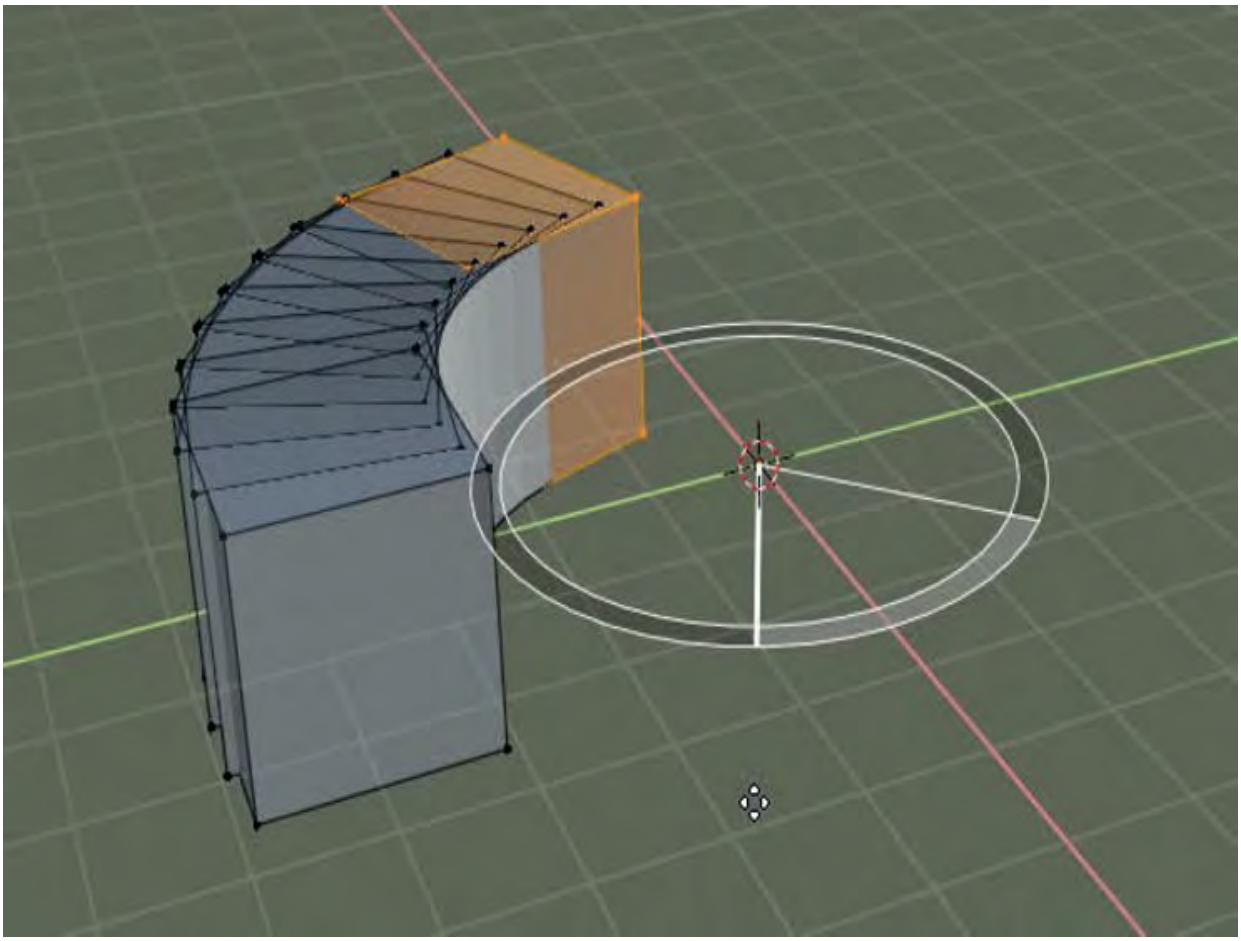


Figure 6.21

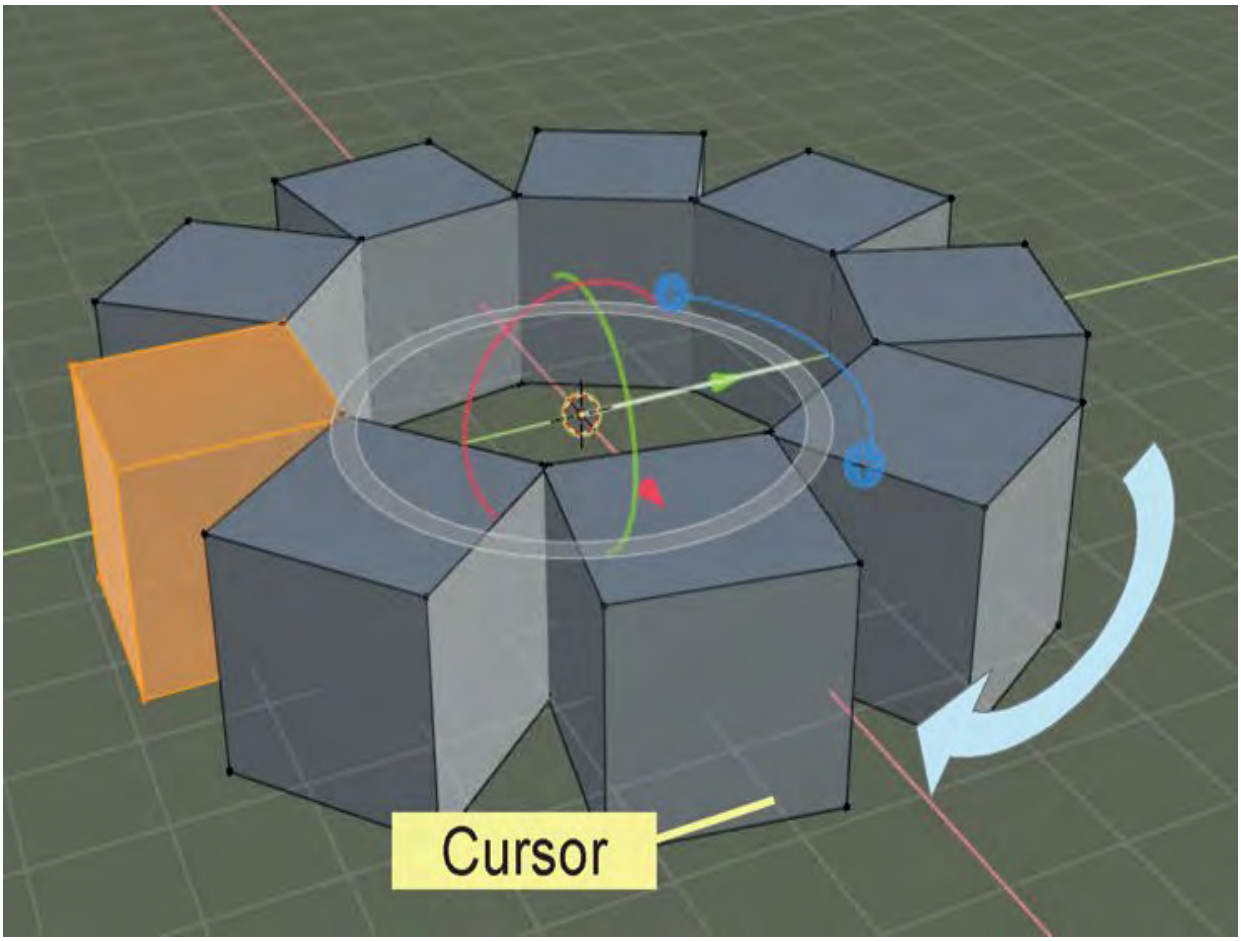


Figure 6.22

Figure 6.23

Release the Mouse button then click on the manipulation Widget that displays and skew the Spin profile (Figure 16.24).

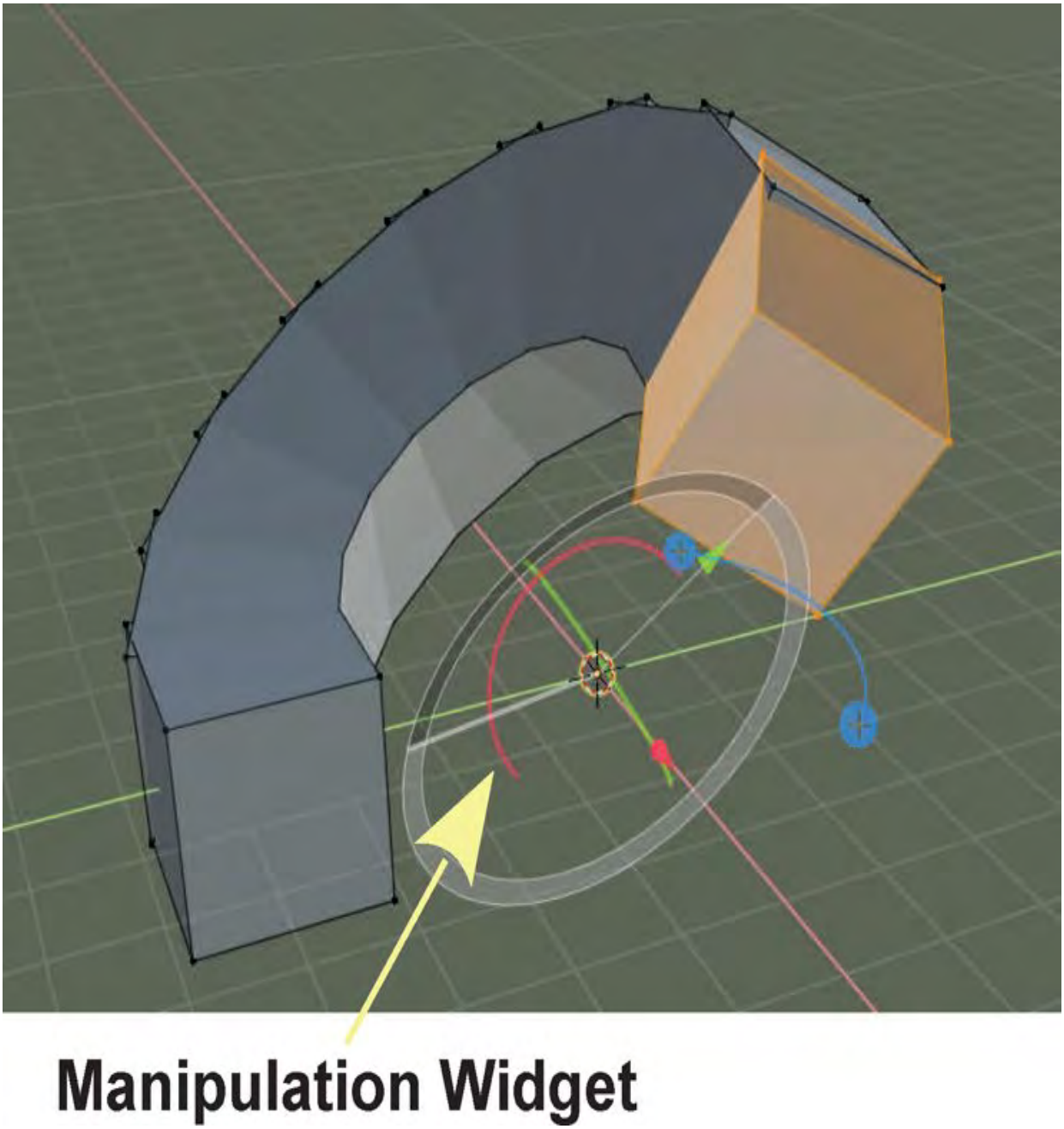


Figure 6.24

6.14 Creating a Spin Profile

A Spin Profile is a line drawing of a shape representing the cross section through an object you wish to generate. As an example a wine glass or goblet will be created.

To create a Spin Profile you start with two vertices in Edit Mode in the 3D Viewport Editor. One way to do this is to have a **Plane Object** in a new Scene and delete two Vertices in Edit mode

A Plane Object is always added to a Scene laying flat in Top Orthographic View, therefore, flick it on edge by pressing **R Key** (Rotate) + **X Key** + **90** (rotate about the X Axis 90 degrees) press **Enter.**

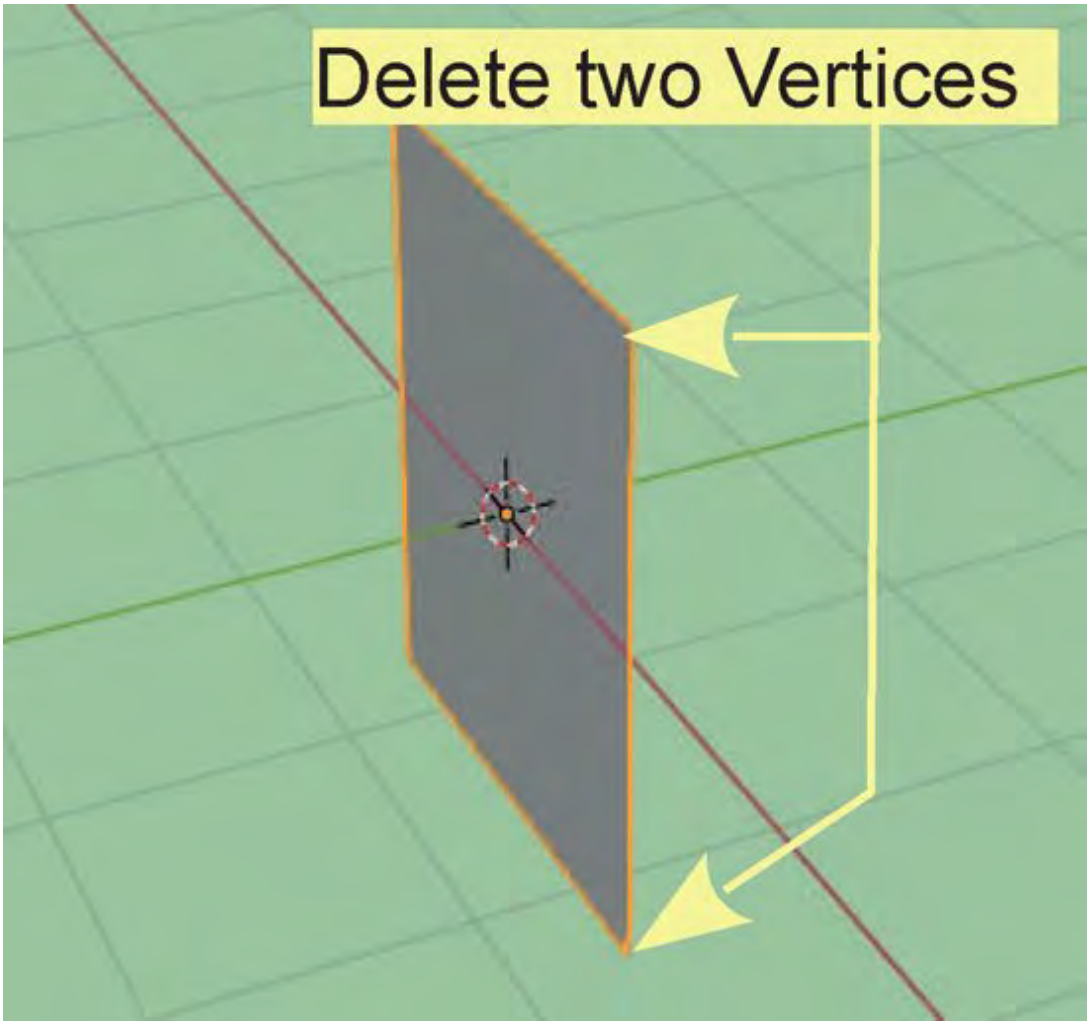


Figure 6.25

Have the Plane in **Front Orthographic View**, in **Edit Mode** and delete two Vertices. This leaves the remaining two Vertices from which build your

profile.

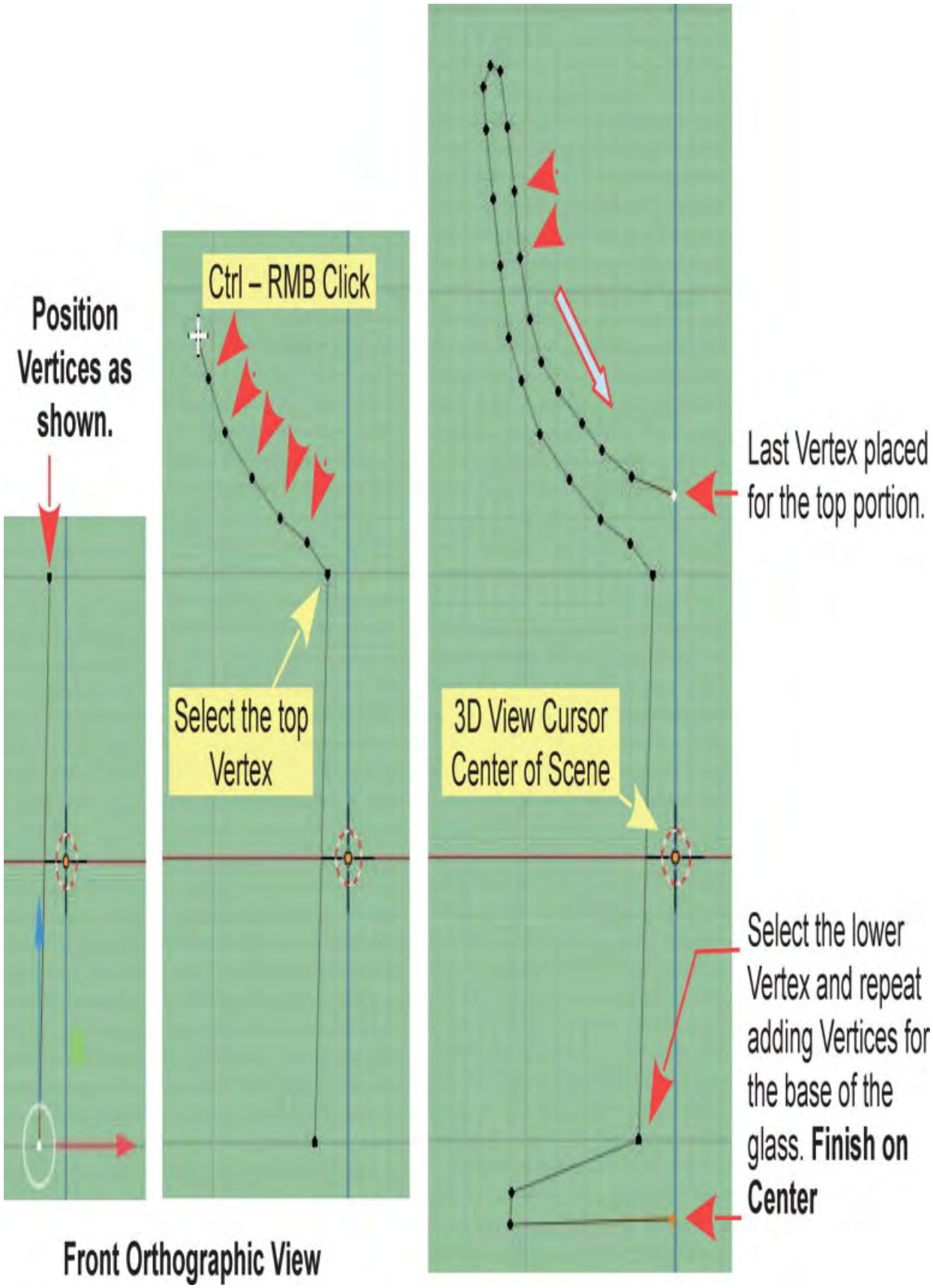

Position Vertices as shown.
Ctrl – RMB Click
Select the top Vertex
3D View Cursor Center of Scene
Last Vertex placed for the top portion.
Select the lower Vertex and repeat adding Vertices for the base of the glass. Finish on Center
Front Orthographic View

Figure 6.26

Spinning: To generate the glass the completed profile will be spun around the position of the **3D View Editor Cursor** with the **3D View Editor** in **Top Orthographic View** in **Edit Mode**. In creating the profile the 3D View Editor has been in **Front Orthographic View** and **Note: The 3D View Editor Cursor is at the center of the Scene.**

Important: If the 3D View Editor Cursor is located other than at the center of the Scene, spin rotation will take place about that point with unexpected results.

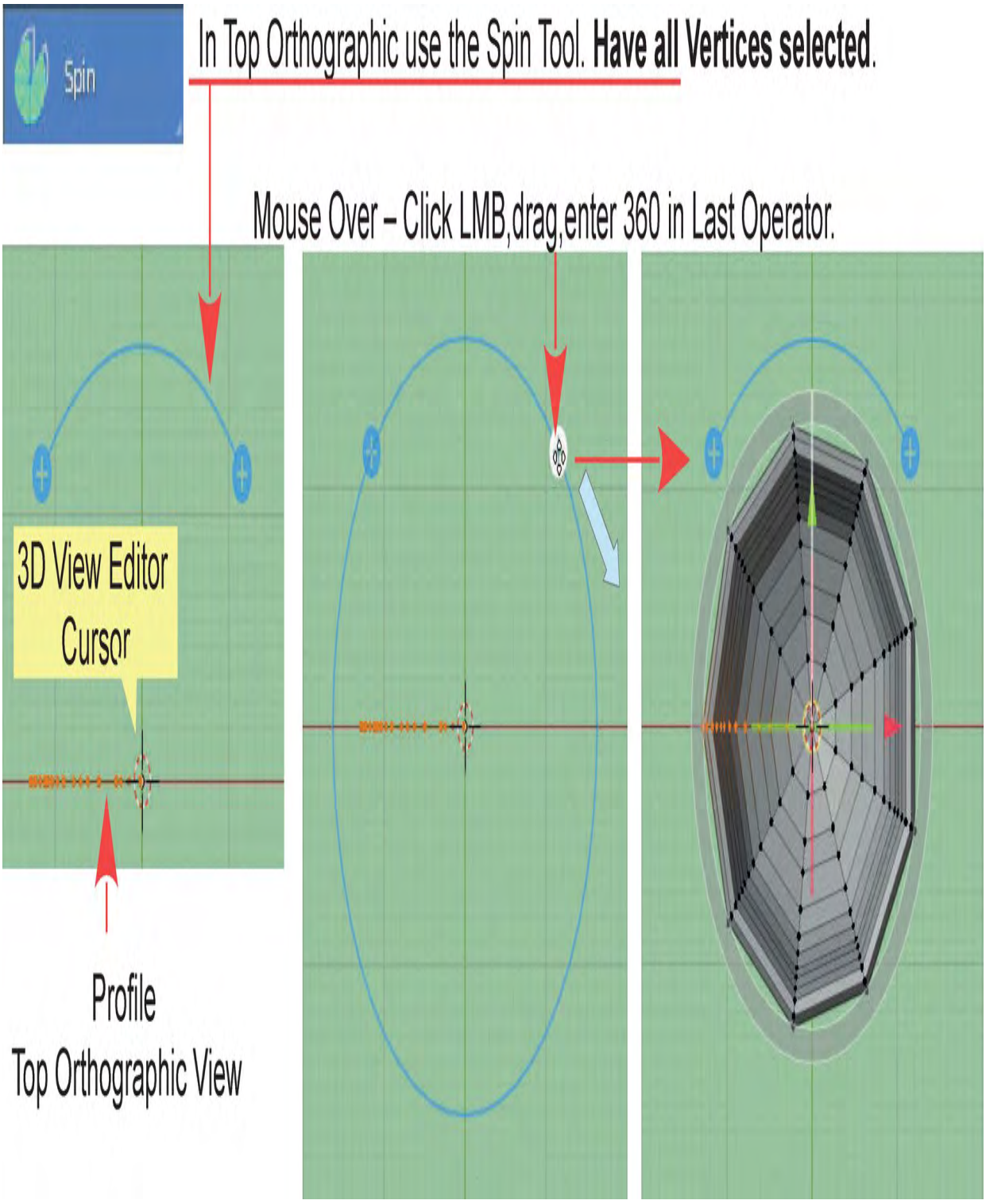
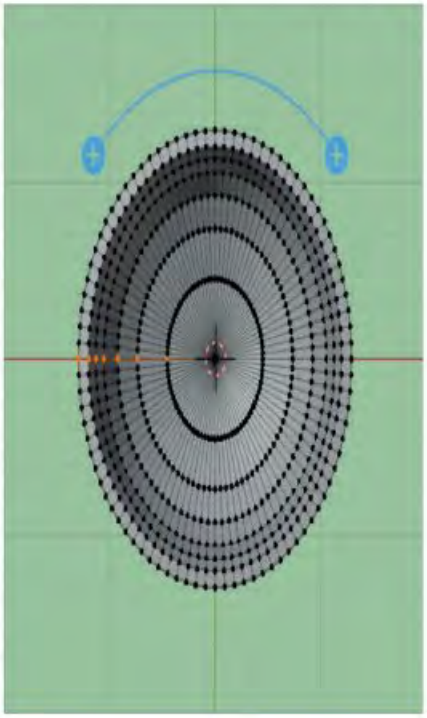
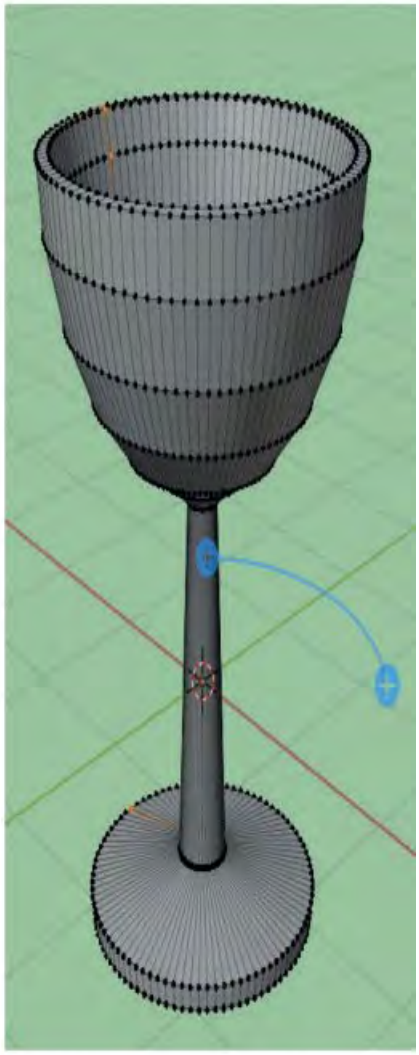

Spin
In Top Orthographic use the Spin Tool. Have all Vertices selected.
Mouse Over – Click LMB,drag,enter 360 in Last Operator.
3D View Editor Cursor
Profile
Top Orthographic View

Figure 6.27

In the **Last Operator Spin Panel** increase the number of **Steps** to add Vertices create a rounding effect.





User Perspective View

In Object Mode, LMB click and select **Shade Smooth** in the **Object Context Menu.**

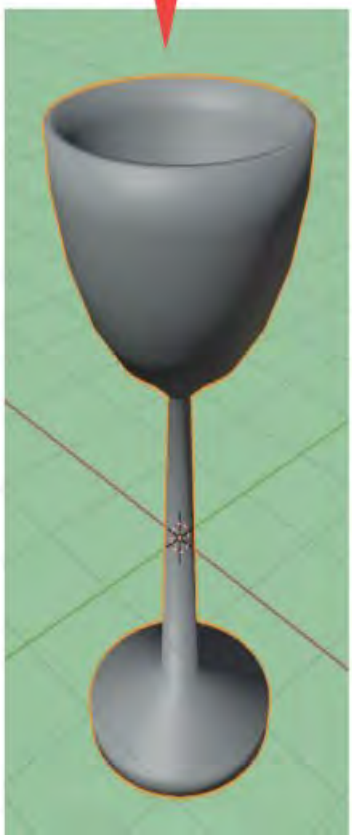


Figure 6.28

6.15 Spin Duplication

The Spin Tool may be used to duplicate an Object around a circular path. The Spin is always about the Z Axis on the XY Plane, therefore, place the 3D Viewport Editor in **Top Orthographic View**. The Spin uses the location of the **3D Viewport Editor Cursor** as the center of rotation.

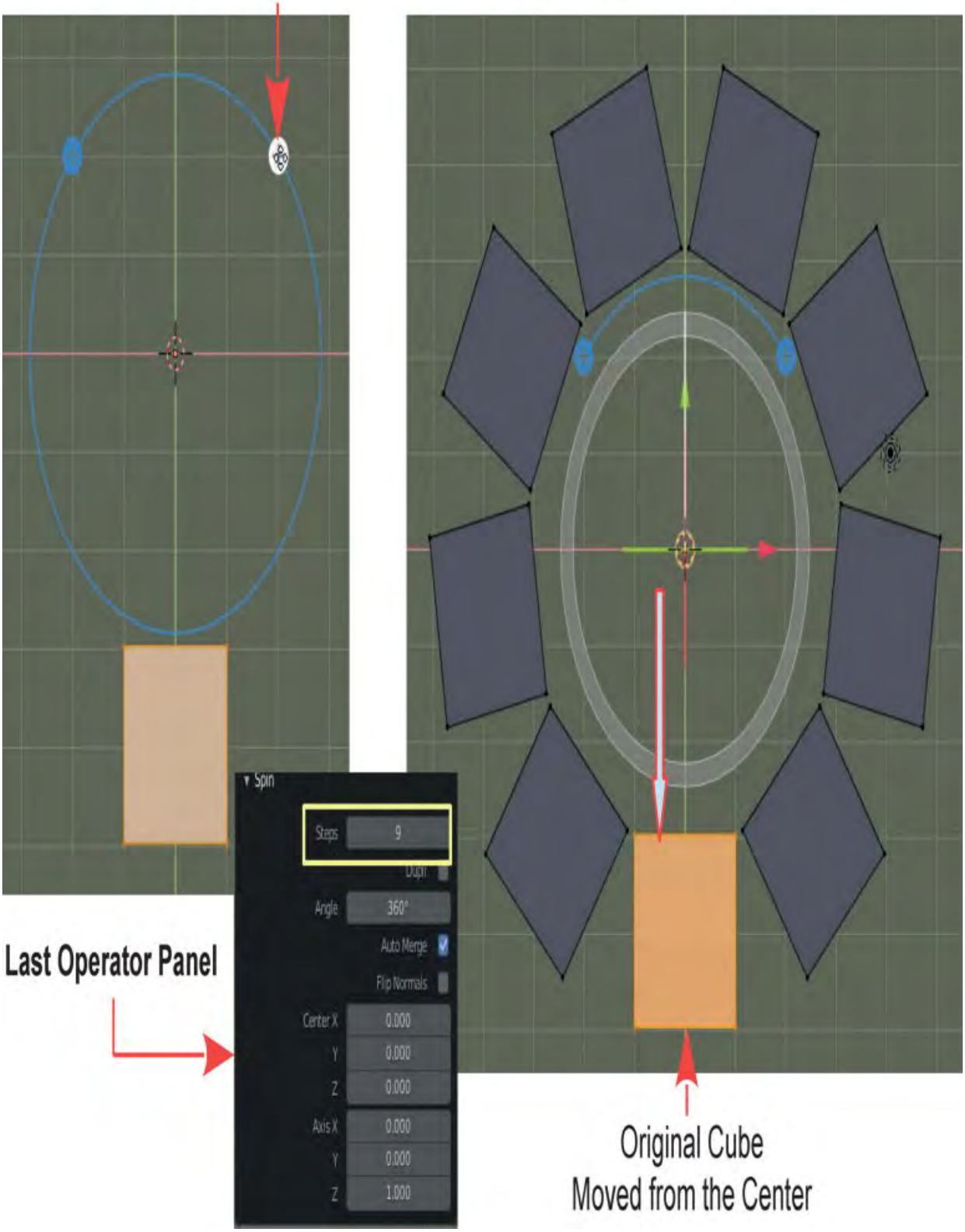
In a new Scene in **Top Orthographic View** with the default **Cube Object** selected and located at the centre of the Scene, Tab to **Edit Mode**. Place the 3D Viewport Editor Cursor off to one side (Shift + RMB click to place). Alternatively, with all Vertices selected move the Cube away from the Cursor which is at the center of the Scene. Activate the **Spin Tools** and click **LMB, hold and drag** on either of the crosses in the blue circles that are displayed. (Figure 6.28).

By default the Spin creates twelve duplications of the Cube spaced in a circle around the 3D Viewport Editor Cursor. The number of duplications depends on the value set in the Last Operator Panel for the Spin (Figure 6.29).

The Vertices of the original Cube are selected (in Edit Mode). You may press the G Key and move the Vertices but if you Tab to Object Mode you will discover that although you appear have separate Cubes, they are in fact joined as one Object. Tab back to Edit Mode. Press the **P Key** and select **Separate by Loose Parts**. In Object Mode you may select individual Cubes and move them but note where the center of each Cube is located. With one Cube selected Tab to Edit Mode and reposition the individual Vertices of the Cube over the center point.

Top Orthographic View
Click LMB on a Blue Cross

Eight Duplication
Nine Steps in the Last Operator Panel



Last Operator Panel

Original Cube
Moved from the Center

Figure 6.29

6.16 The Screw Tool

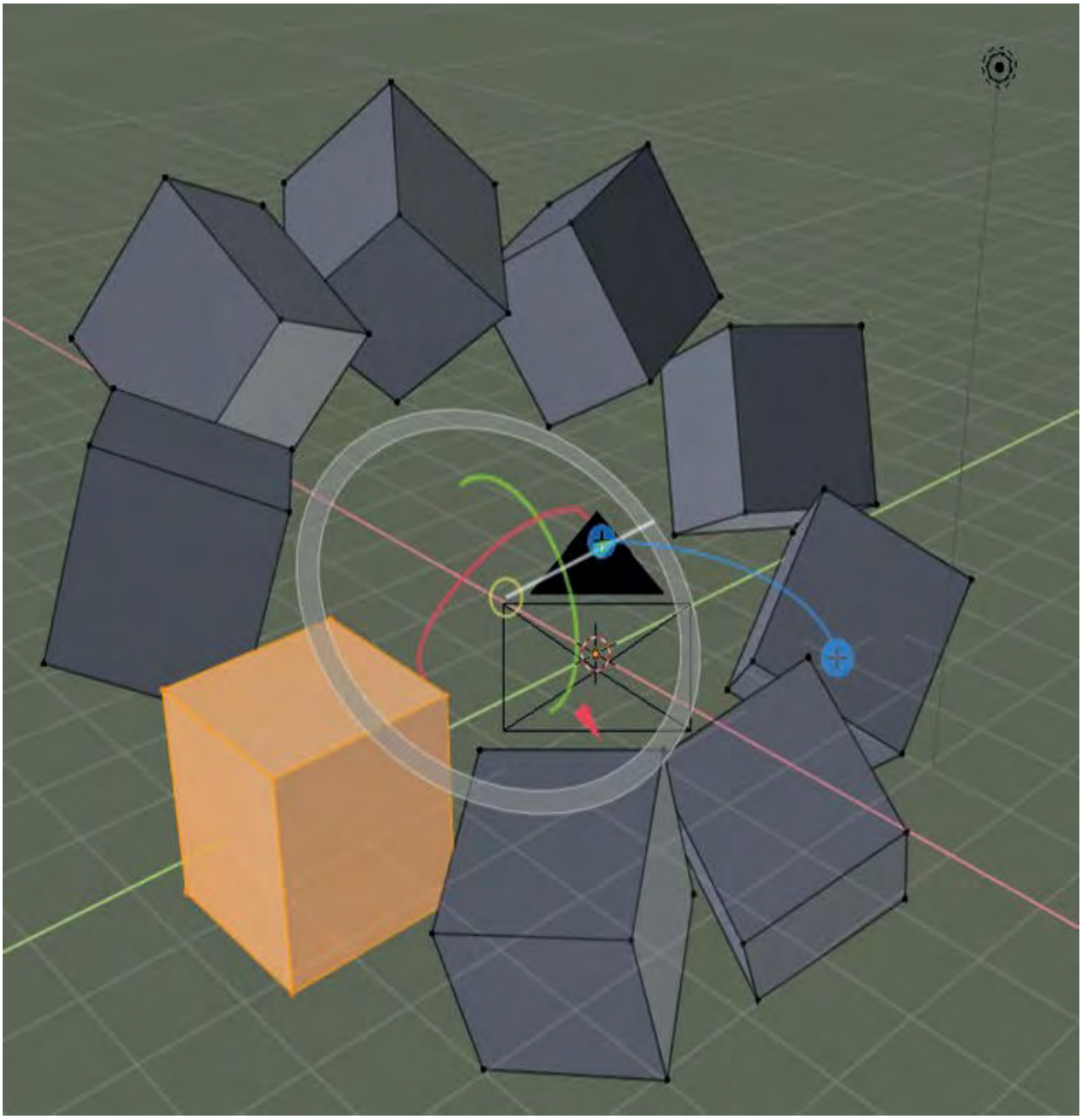


Figure 6.30

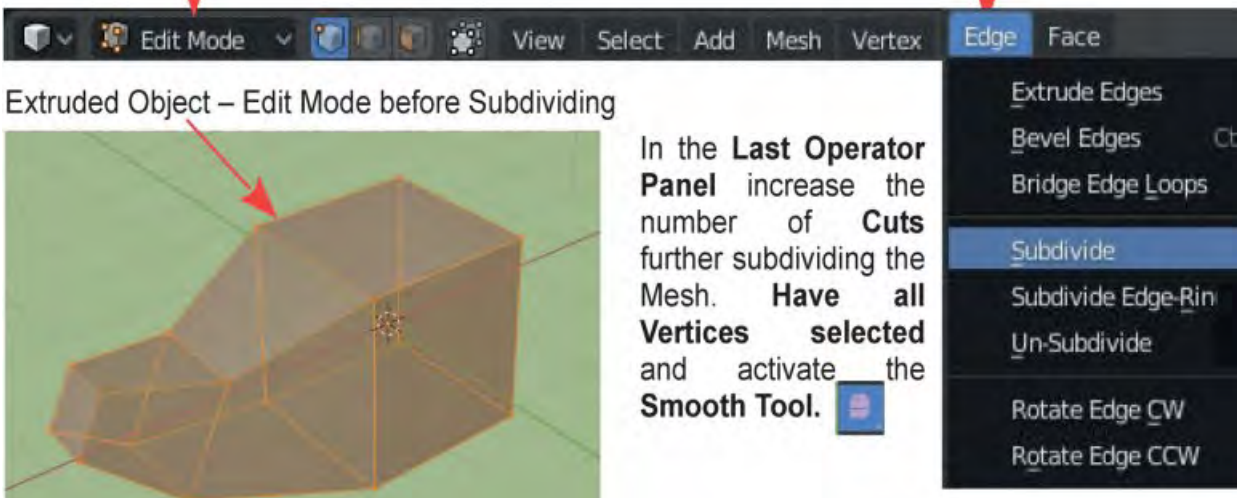
Previous versions of Blender included a Screw Tool which spun vertices and at the same time generated duplications at right angles to the spin

producing a Screw effect. This procedure is now accomplished by the **Screw Modifier** (see Modifiers Chapter 7). You may, however, use the Manipulation Widget of the Spin Tool to skew the Spin Duplication.

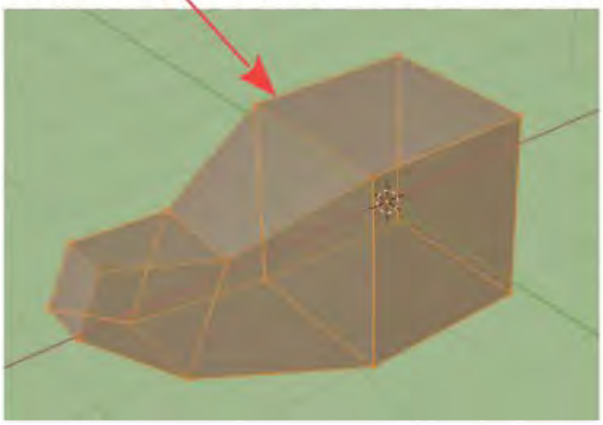
6.17 The Smooth and Randomize Tools

The effect of the **Smooth Tool** is to smooth or round transitions at the corners of a mesh. Randomize displaces Vertices in a Random order. The Tools are best demonstrated with an Object that has been Extruded and Subdivided (see Section 6.4).

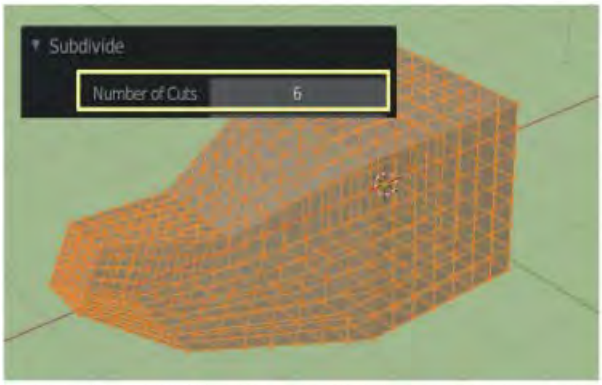
In the **3D Viewport Editor Header** select **Edge** and click **Subdivide with all Vertices selected**.



Extruded Object – Edit Mode before Subdividing



In the **Last Operator Panel** increase the number of **Cuts** further subdividing the Mesh. **Have all Vertices selected** and activate the **Smooth Tool.**



In the Smooth Vertices Panel increase Smoothing to 1.000.

After Subdividing and Smoothing

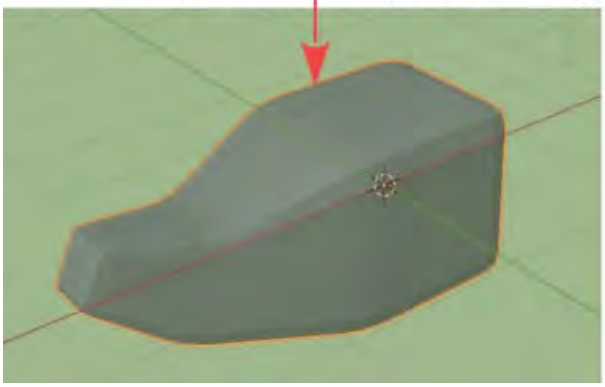
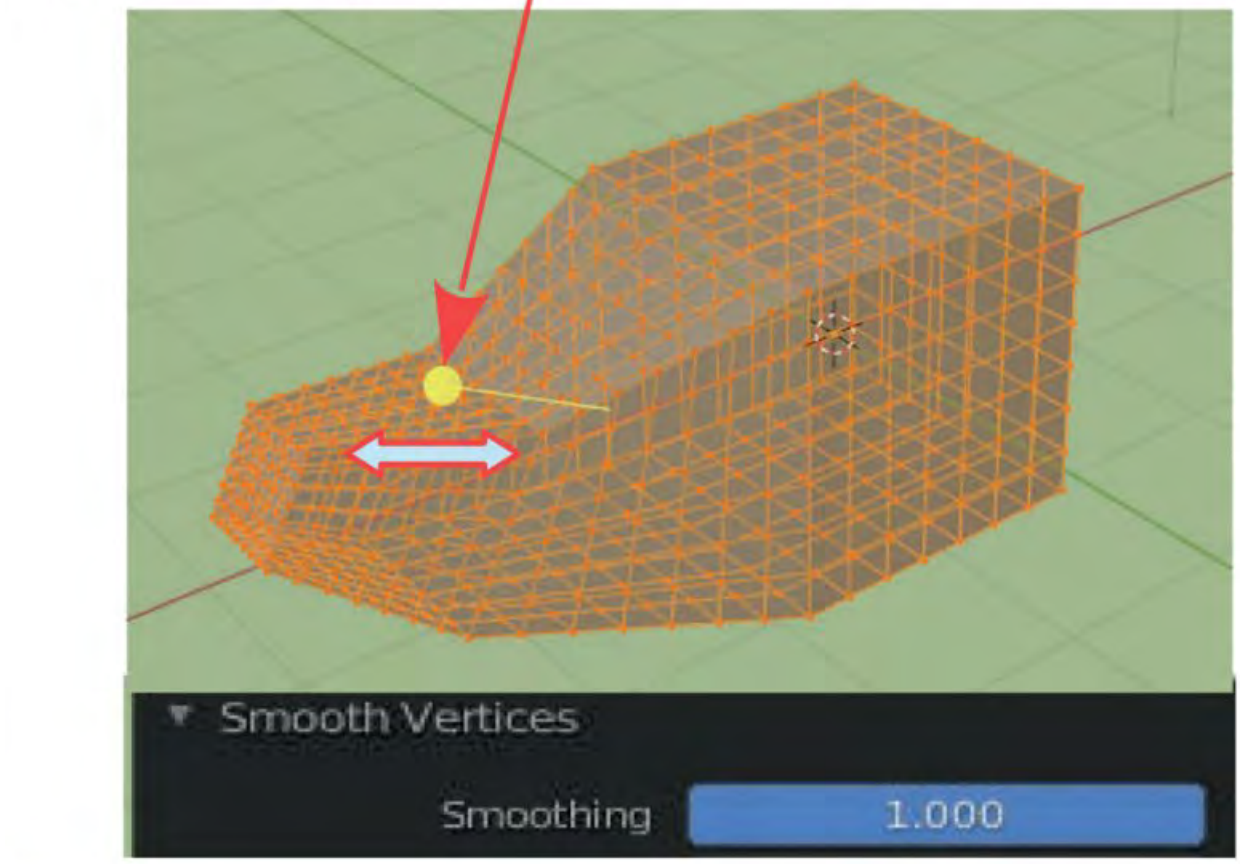


Figure 6.31

LMB Click on the yellow paddle that displays, hold and drag.



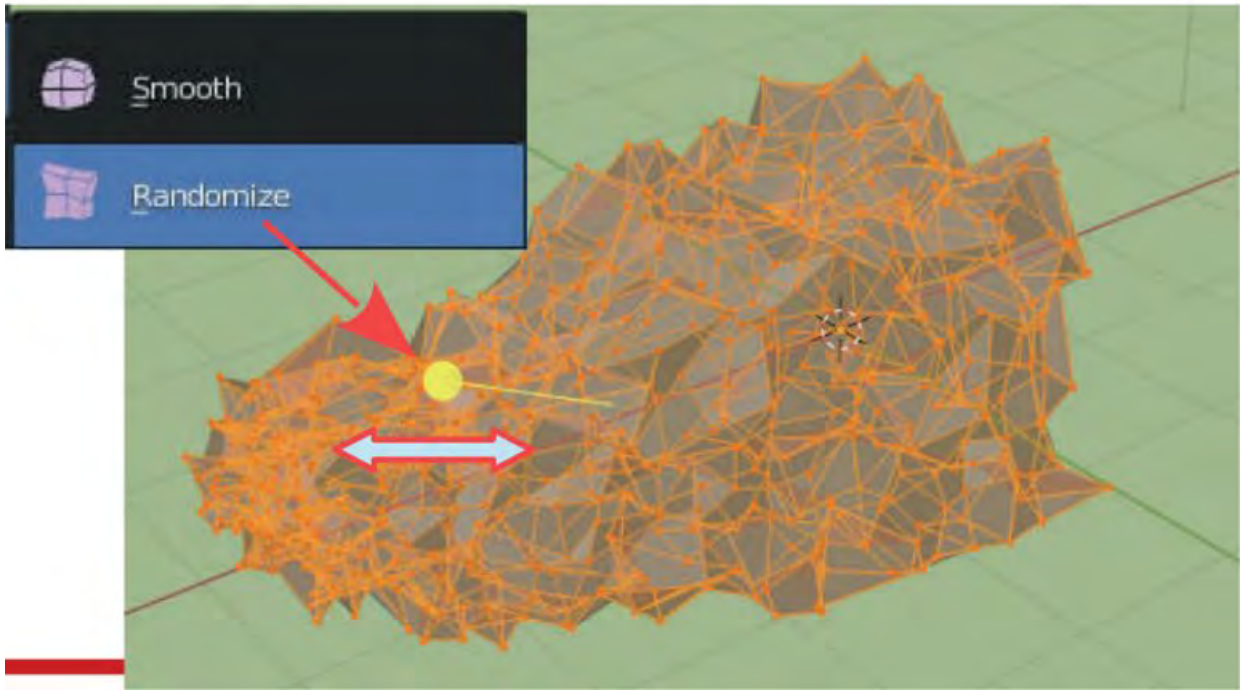


Figure 6.32

6.18 The Edge Slide And Vertex Slide Tool

Select an **Edge** (in Edit Mode) on an Object. Activate the **Edge Slide Tool**. Click on the yellow paddle and drag the Mouse. The Edge will be translated within the space occupied by the Object.

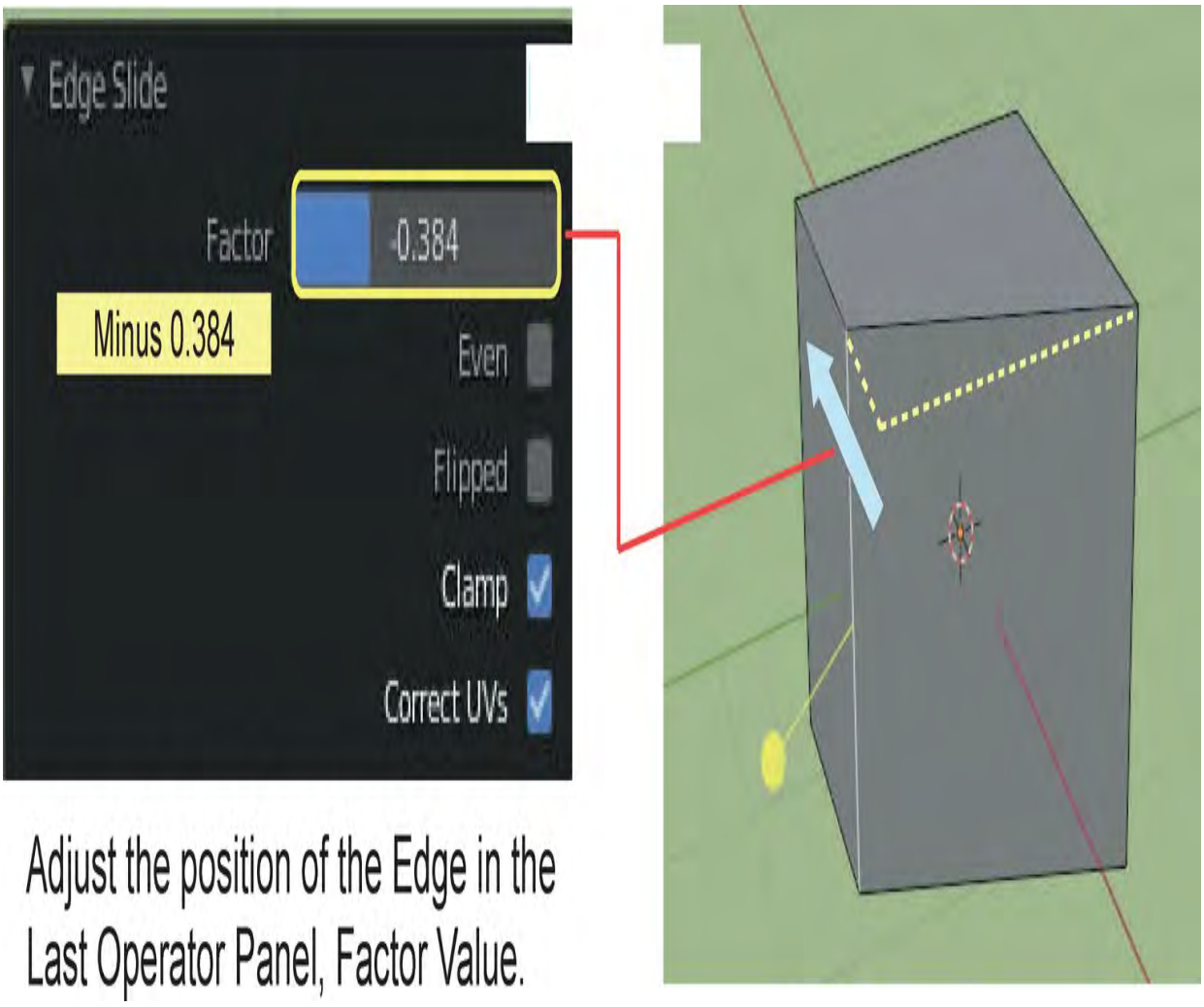


Figure 6.33

6.19 The Shrink Fatten Tool

The **Shrink Fatten** Tool allows you to expand (fatten) or shrink a selection.

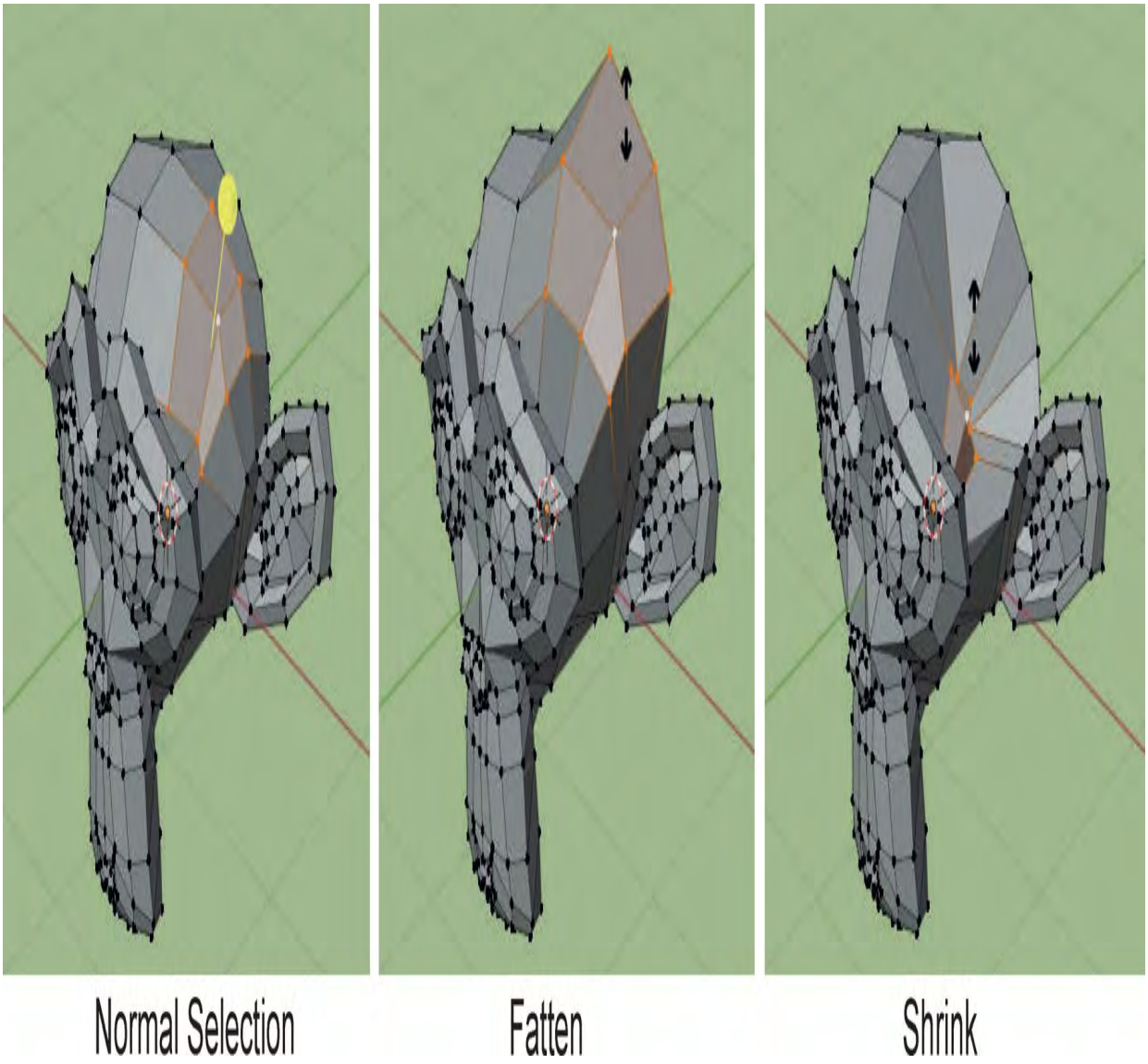


Figure 6.34

6.20 The Shear Tool

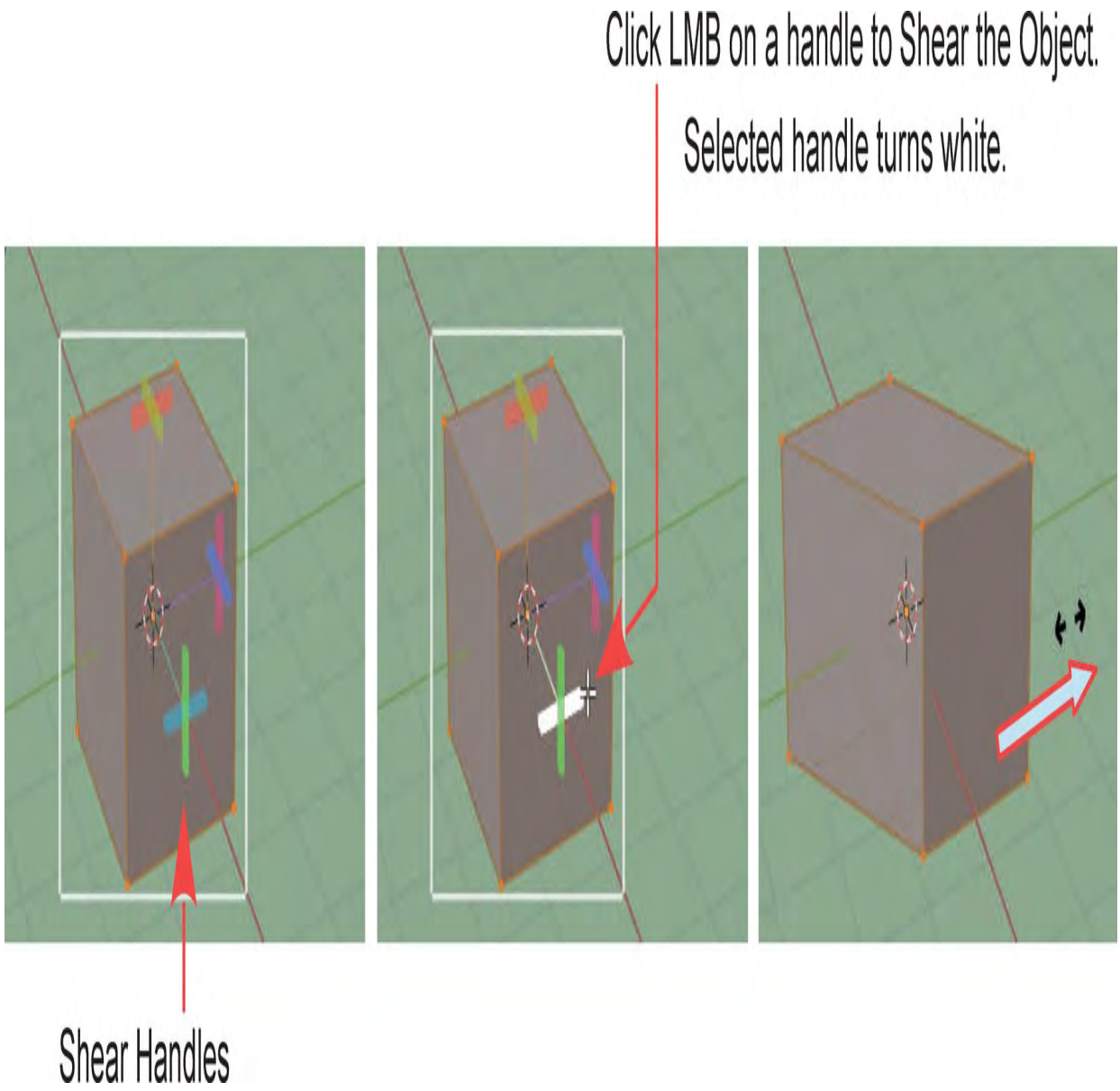


Figure 6.35

6.21 The Rip Region Tool

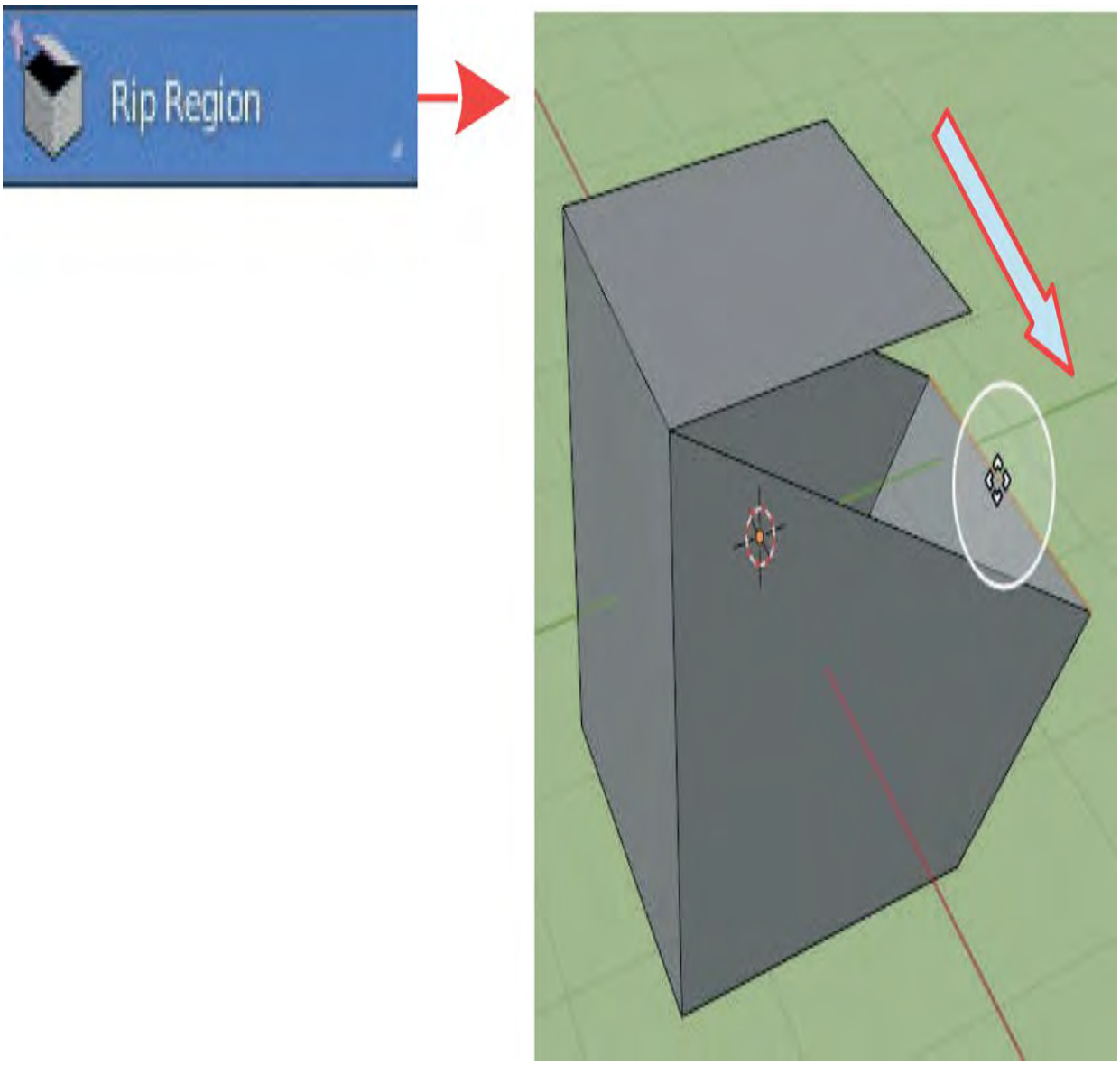


Figure 6.36

Select an Edge (Edit Mode – Edge Selection Mode). Activate the **Rip Region Tool**. With the Mouse Cursor in the Editor, click, hold and drag to translate the selected Edge.

See also; The Edge Split Modifier Chapter 8 – 8.6 for performing similar operations.

7 Modifiers

7.1 Modifiers in General
7.2 The Modifier Stack
7.3 The Modifier Group
7.4 The Simulate Group
7.5 Generate and Deform Modifiers

Modifiers, in Blender, are pre-assembled code that apply a process or algorithm to an Object, changing the Object's properties and affecting the way the Object behaves or how it is displayed. Modifiers may, therefore, be considered as Editing Tools.

The Modifiers are designed to automate some of the otherwise tedious processes involved in shaping Objects and controlling their behaviour. Some Modifiers can only be used in conjunction with other processes.

The following chapters are offered as a guide. You will be shown how a Modifier is added to an Object and provided with examples showing the Modifier's basic features.

Modifiers are found in the **Properties Editor, Modifier Properties**.

The Object Modifier Properties are only displayed when an object to which a Modifier can be applied is selected in the 3D Viewport Editor. Some objects can not have Modifiers applied.

Note: If there are Objects in the 3D Viewport Editor to which Modifiers may be applied (not necessarily selected), clicking the **Add Modifier** button and selecting a Modifier will apply the Modifier to the last Object

that was selected. This occurs even though that Object is not selected at the time.

7.1 Modifiers in General

Modifiers are found in the **Properties Editor, Modifiers Properties.**

Note: The Modifiers Properties button is only displayed when an Object to which a Modifier can be applied is selected in the 3D Viewport Editor. Some Objects cannot have Modifiers applied.

With an Object selected in the 3D Viewport Editor (in Object Mode), clicking the Modifier Properties button in the Properties Editor displays the empty panel shown in Figure 7.1.

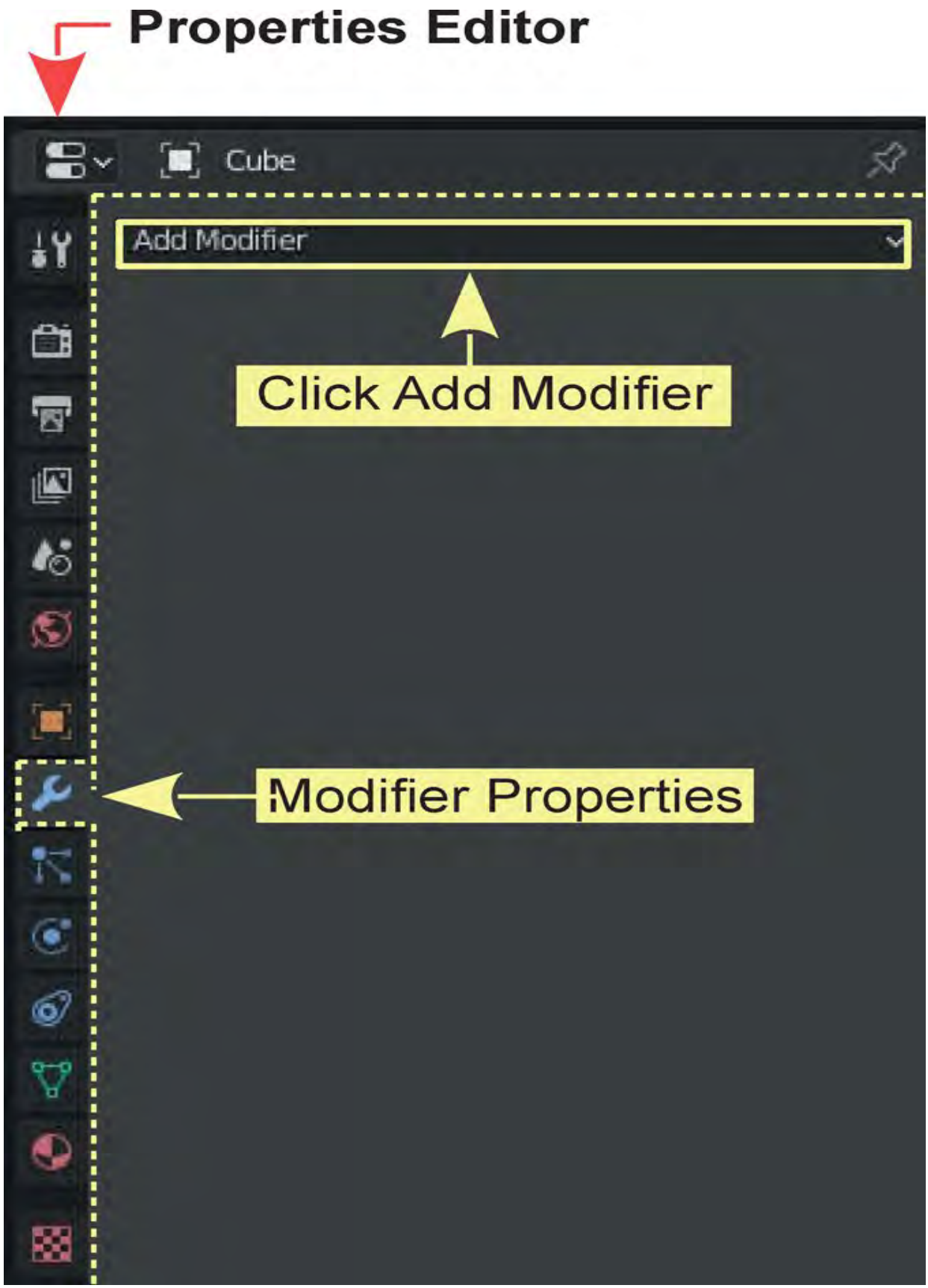

Properties Editor
Cube
Add Modifier
Click Add Modifier
Modifier Properties

Figure 7.1

In this chapter **Modifiers in general** will be briefly described. Some Modifiers are complex and beyond a basic introduction and will be better understood having undergone further studies. New Modifiers are continually being added to the program and the selection menu changes accordingly. Click on the **Add Modifier** button to view the menu (Figure 7.2).

Modify
Data Transfer
Mesh Cache
Mesh Sequence Cache
Normal Edit
Weighted Normal
UV Project
UV Warp
Vertex Weight Edit
Vertex Weight Mix
Vertex Weight Proximity
Generate
Array
Bevel
Boolean
Build
Decimate
Edge Split
Mask
Mirror
Multiresolution
Remesh
Screw
Skin
Solidify
Subdivision Surface
Triangulate
Weld
Wireframe
Deform
Armature
Cast
Curve
Displace
Hook
Laplacian Deform
Lattice
Mesh Deform
Shrinkwrap
Simple Deform
Smooth
Smooth Corrective
Smooth Laplacian
Surface Deform
Warp
Wave
Simulate
Cloth
Collision
Dynamic Paint
Explode
Fluid
Ocean
Particle Instance
Particle System
Soft Body

Figure 7.2

The menu is divided into four categories to aid selection.

To demonstrate the basic procedure for adding a **Modifier** the **Generate – Bevel Modifier** will be used. Begin by having the default Cube Object selected in the **3D Viewport Editor**. In the **Properties Editor, Modifier Properties**, click the **Add Modifier** button to display the selection menu (Figure 7.2).

> This instruction has been purposely repeated

Under the **Generate** heading select (click LMB) **Bevel**.

The **Bevel Modifier panel** opens in the **Properties Editor** and the Cube object in the 3D Viewport Editor displays with its edges bevelled (Figure 7.3). Values in the Modifier panel may be adjusted to affect the bevel.

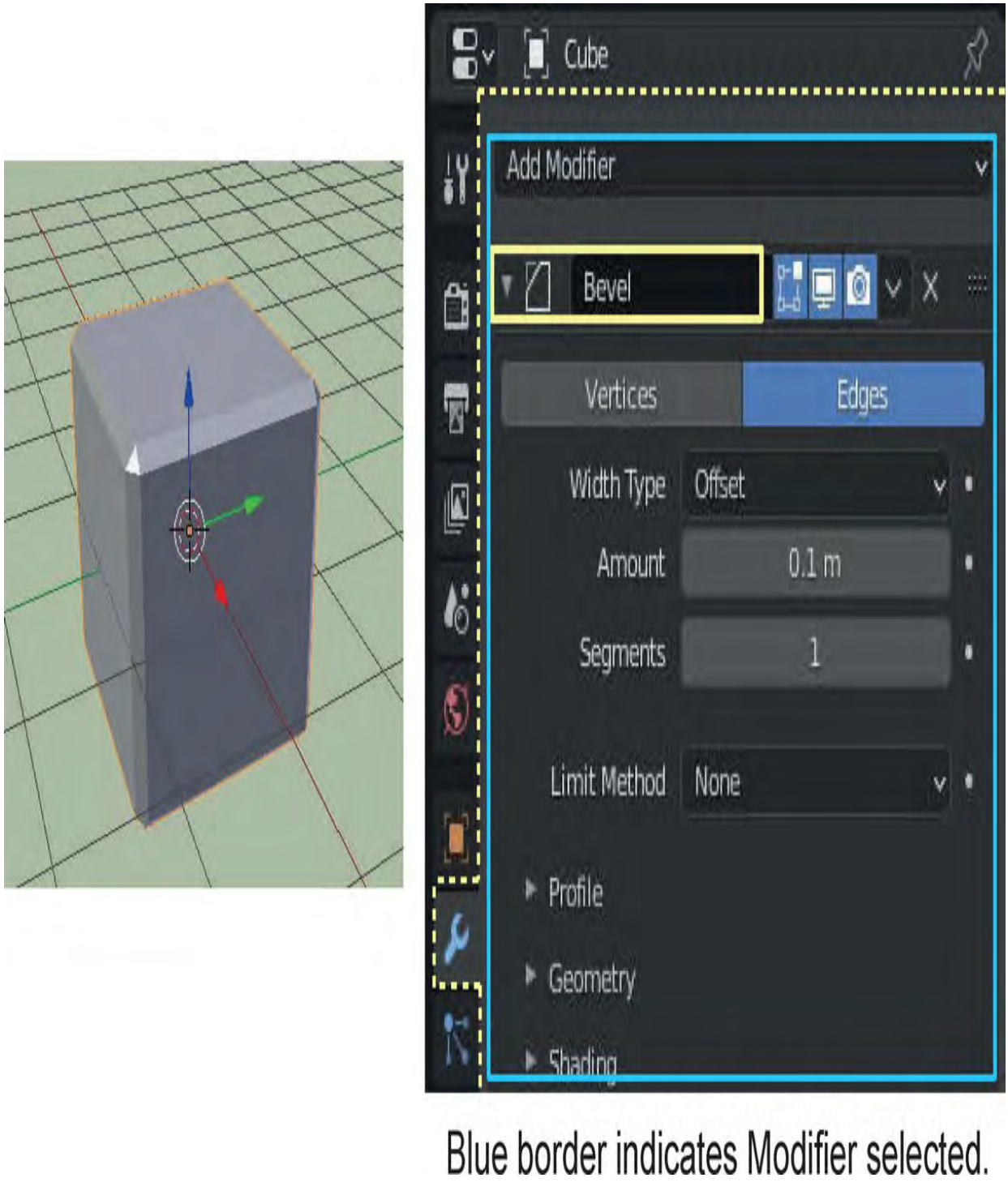


Figure 7.3

By default, when panels are opened in Blender, they tend to open with a minimal display to save space in the interface. The Modifier panel is no

exception.

Before attempting an exercise in using a Modifier you should understand a few basic features of the typical Modifier panel. Keep in mind that each type of Modifier will display a different panel configuration. The bevel Modifier panel will be used as an example.

Figure 7.4 shows the upper part of the Modifier panel which is typical of all Modifiers.

The selected Object in the 3D Viewport Editor
Modifier Type (Name)
Toggle Show / Hide the Modifier panel
Delete the Modifier
Cube
Add Modifier
Bevel
Edit Mode.
Display modifier in Edit mode.
Realtime.
Display modifier in viewport.
Render.
Use modifier during render.
Apply
Ctrl A
Duplicate
Shift D
Move to First
Move to Last
See over page for further detail.

Figure 7.4

See over page for further detail.

Edit Mode (Display modifier in Edit Mode):

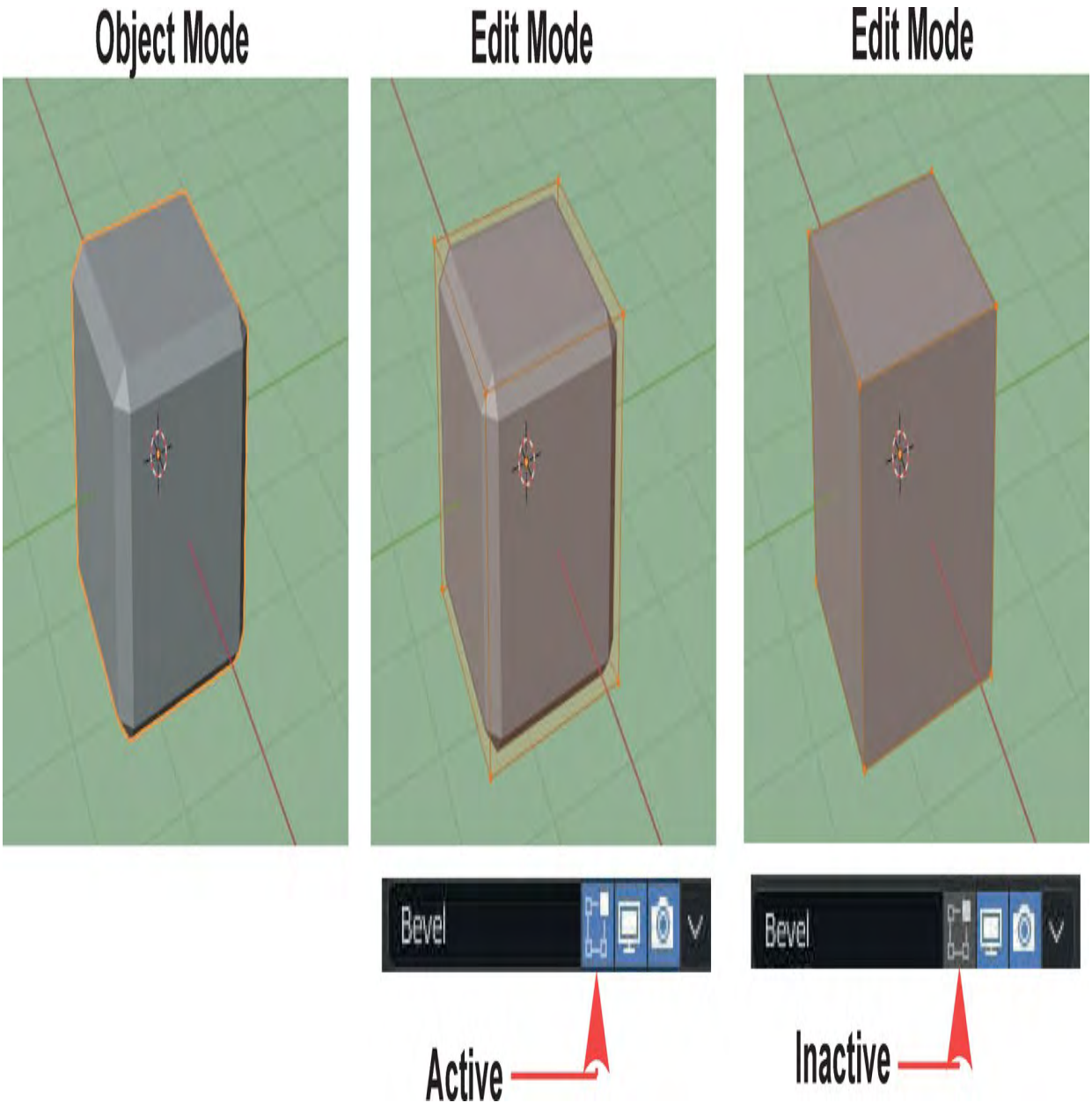


Figure 7.5

Realtime (Display Modifier in Viewport)

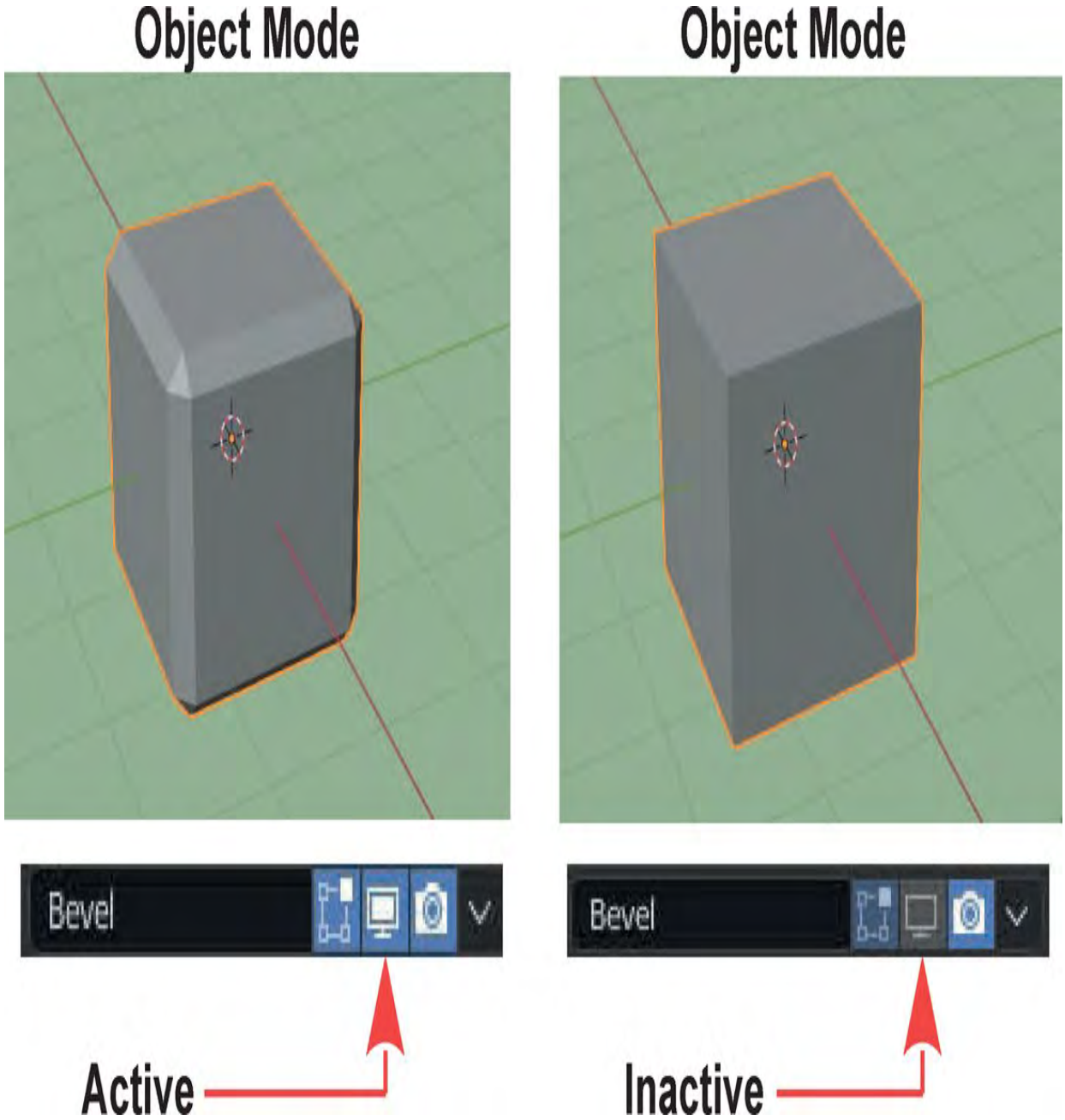


Figure 7.6

Render (Use Modifier during Render)

Pressing the **F12** Key Renders a view of Camera View

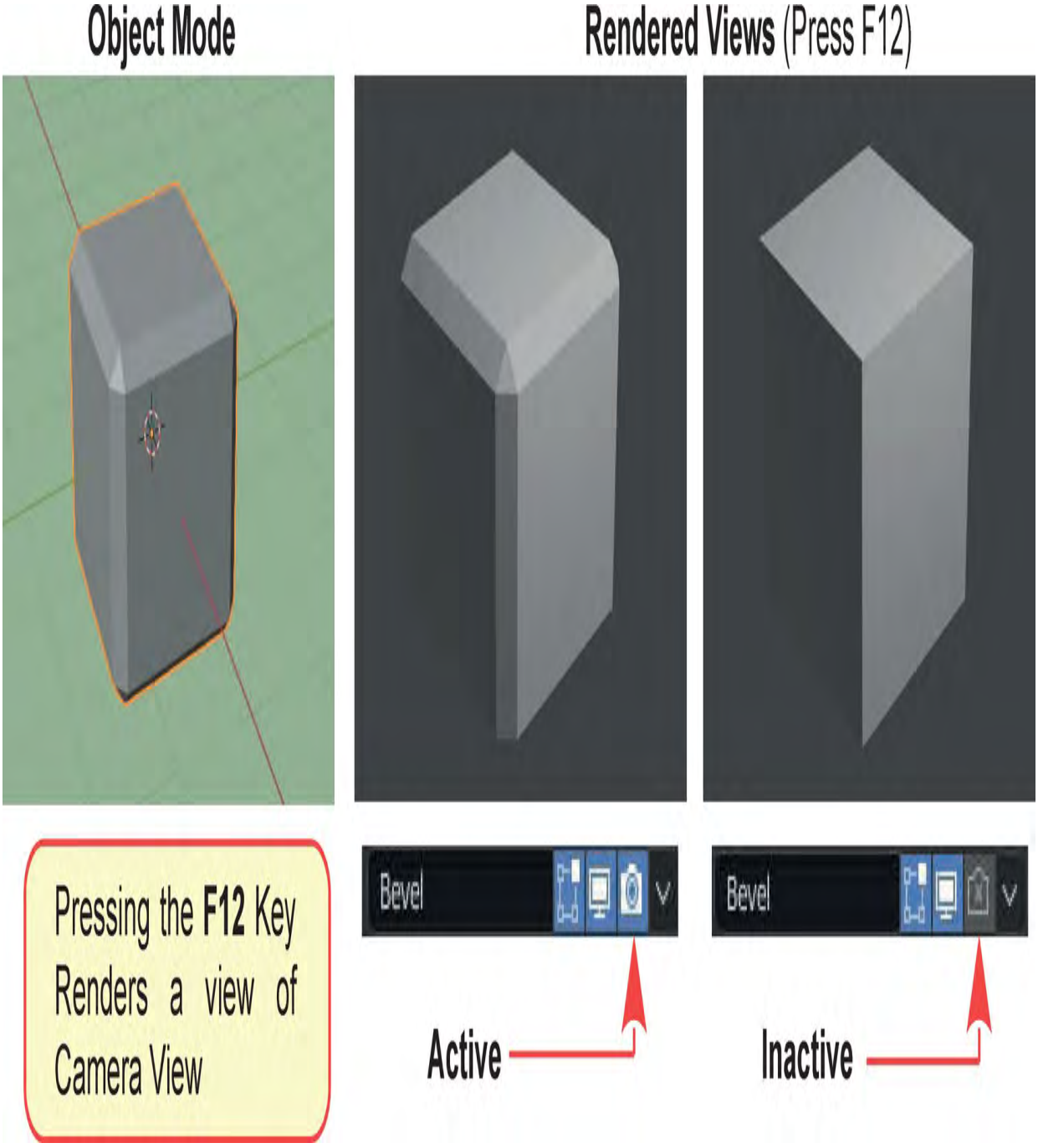


Figure 7.7

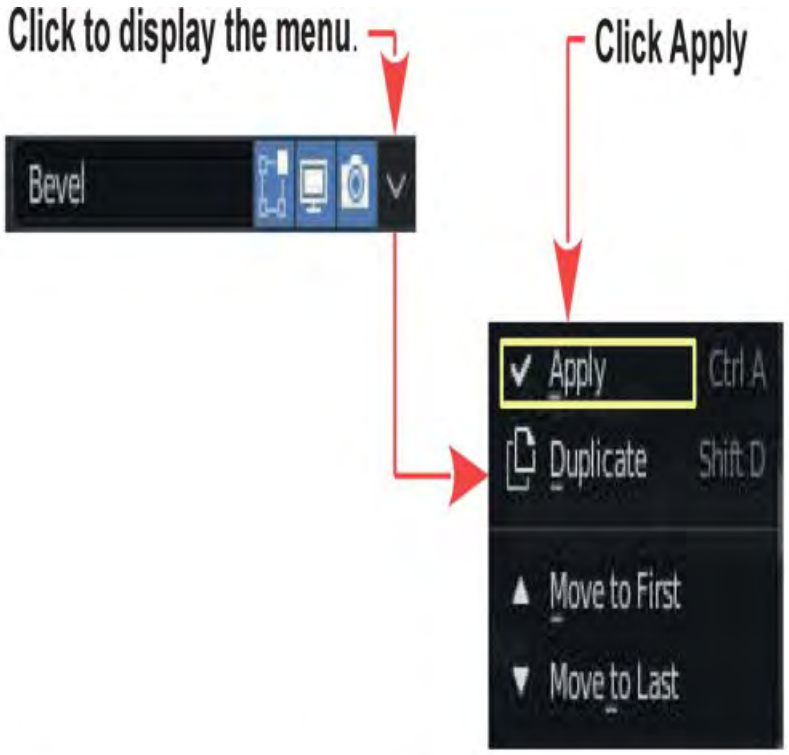
When adjusting values **in the Modifier Panel** you should be aware of the **Apply button.**

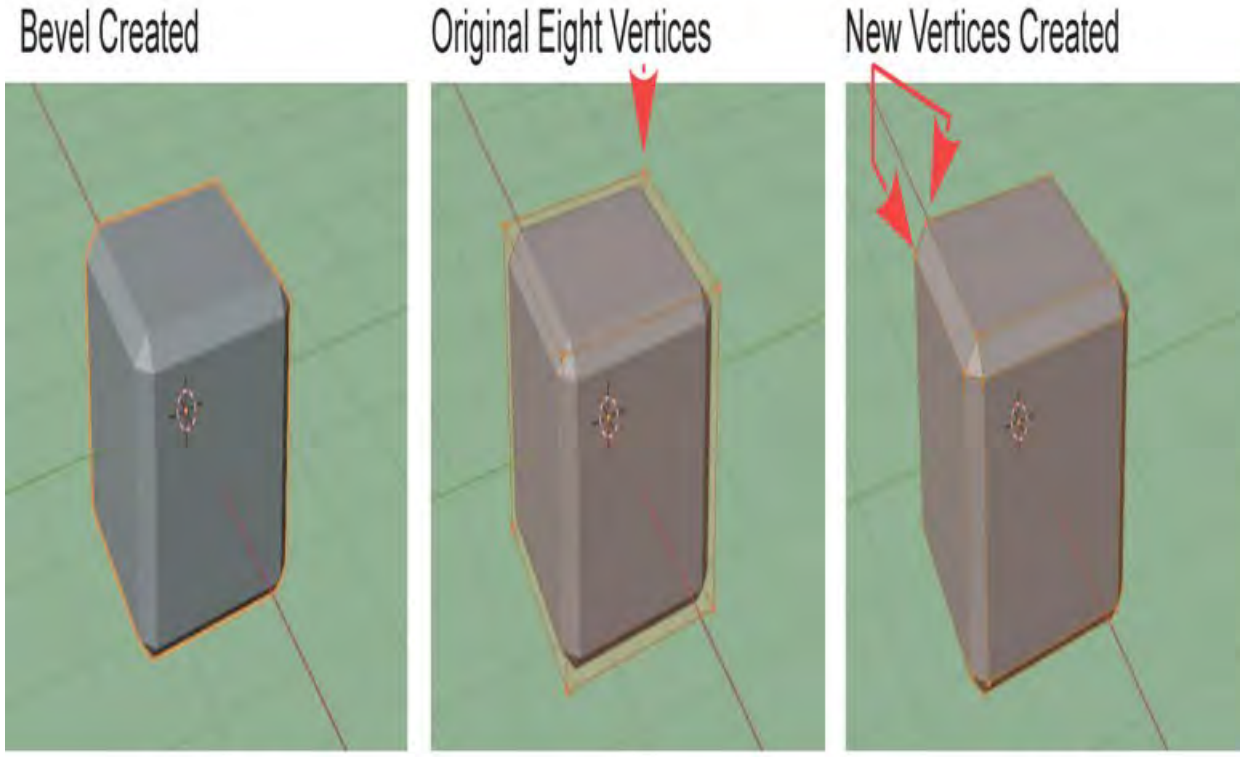
Adding and Applying Modifiers

At this point how to Add a Modifier to an Object has been described. You should understand that using a Modifier entails two stages. The first stage is to Add the Modifier to the Object and adjust values in the Modifier panel. The second stage is to Apply the Modifier. Applying assigns data to the Object permanently affecting the Object.

In the case of the Cube Object, Adding the Bevel Modifier sees the edges of the Cube bevelled in the 3D Viewport Editor when in Object Mode. In Edit Mode, with Display Modifier in Edit Mode Active, you will see the bevelled Edges with the Viewport in Edit Mode but observe the original eight Vertices forming the Cube still exist. The Modifier has been Added but has NOT been Applied.

To Apply the Modifier you click the hidden **Apply** button in the Modifier panel.





Object Mode
Bevel Modifier Added

Edit Mode
Bevel Modifier Added

Edit Mode
Bevel Modifier Applied

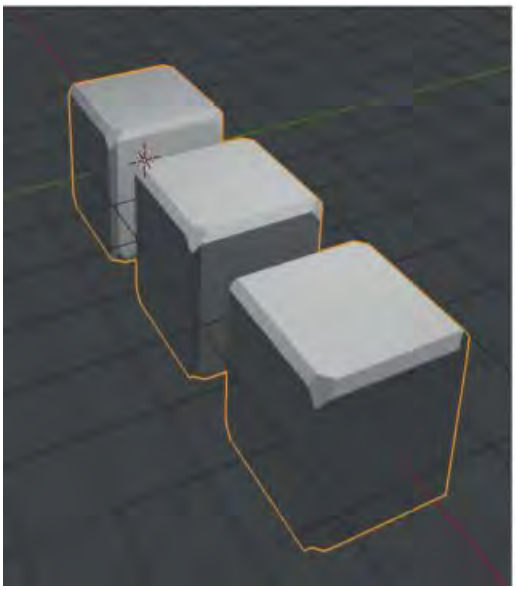
Figure 7.8

Note: Applying the Modifier can only be performed in Object Mode.

Note: You may delete a Modifier **before Applying,** cancelling all the modifications made (see the **Delete** button in Figure 7.4. Once a Modifier has been Applied all modifications are permanently set and the Delete option is no longer available in the Modifier panel. However, if the Modifier has been Applied within the range of the **Undo function** you can press **Ctrl + Z** with the Mouse Cursor in the 3D Viewport Editor and backstep through operations until you reach the point before the Modifier was Added to the Object.

7.2 The Modifier Stack

In some cases it is appropriate to apply more than one Modifier to an Object. The modifiers are placed in a stack in order of priority. A Modifier at the top of the stack takes precedence over Modifiers lower down. The priority can be changed by moving a Modifier up or down in the stack. Although Modifiers are generally applied in Object mode, some may be used in Edit mode. Figure 7.9 shows an **Array Modifier**, a **Bevel Modifier** and a **Weld Modifier** in a stack.



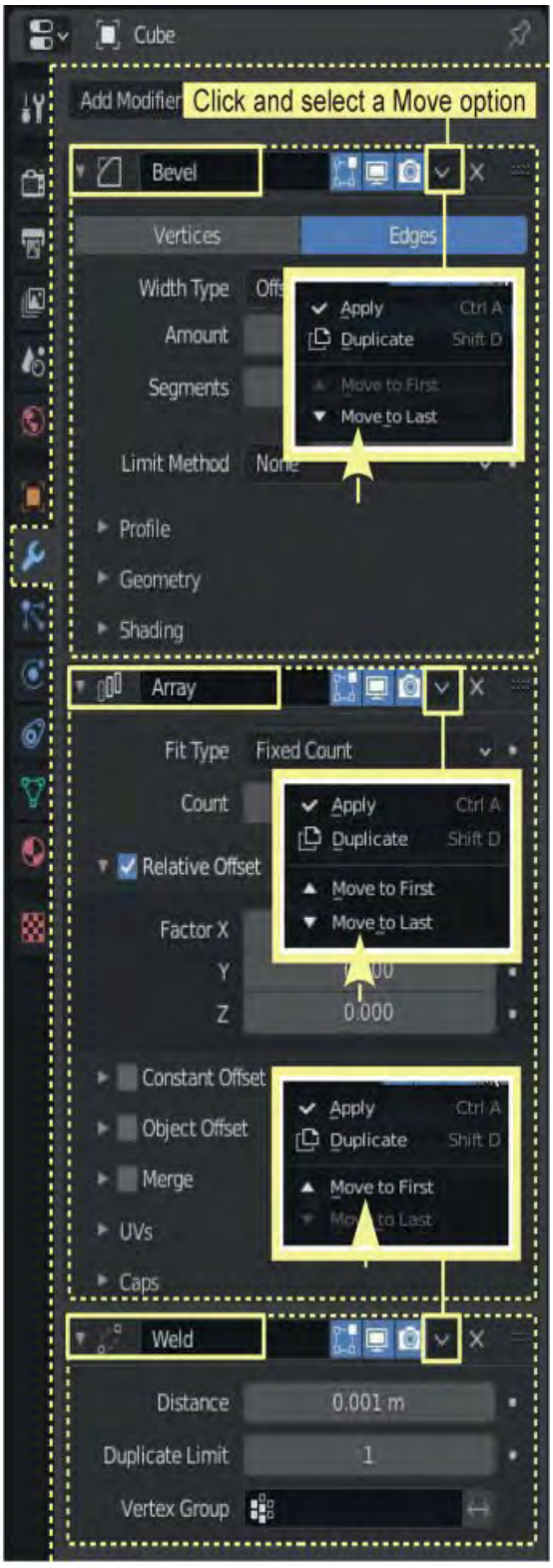

Cube
Add Modifier
Click and select a Move option
Bevel
Vertices
Edges
Width Type
Amount
Segments
Limit Method
Apply Ctrl A
Duplicate Shift D
Move to First
Move to Last
Profile
Geometry
Shading
Array
Fit Type
Fixed Count
Count
Relative Offset
Factor X
Y
Z
0.000
Constant Offset
Object Offset
Merge
UVs
Caps
Weld
Distance
0.001 m
Duplicate Limit
1
Vertex Group

Figure 7.9

The **Bevel Modifier** has been added first followed by the **Array Modifier** then the **Weld Modifier**.

With the Modifiers arranged in this order (Bevel at the Top) the Bevel Modifier takes precedence over the other two.

The priority can be changed by selecting a **Move Option** to move the Modifier up or down in the stack.

The Bevel Modifier bevels the edges of the Cube and the Array modifier duplicates the Cube in the 3D Viewport Editor.

In this particular instance the Weld Modifier is not used and is only shown to demonstrate the Modifier Stack.

In the following pages you will be shown how some of the Modifiers are used. The full listing of Modifiers available are shown in the Modifier selection menu (Figure 7.2).

Where Modifiers are used in conjunction with other processes they will be described in the chapter which relates to that process. For example: Armatures (Chapter 22), Particle Systems (Chapter 24), Fluid Simulation and Smoke Simulation (Chapter 25).

7.3 The Modify Group

The Modifiers under the heading **Modify** (Figure 7.2) do not directly affect the shape of an object but rather other data such as Vertex Groups and appearance. The demonstration of this group will be left in abeyance at this point since you will have to study some of Blender's more advanced features before being able to understand what they do.

7.4 The Simulate Group

The **Simulate** group of modifiers, activate simulations (animated physical sequences such as fluid or gas flow). In most cases, these modifiers are

automatically added to the Modifiers stack whenever a **Particle System** or **Physics Simulation** is enabled. Their role is to define the place in the modifier stack used as base data by the tool they represent. Generally, the attributes of these modifiers are accessible in separate panels.

This group of Modifiers is discussed in Chapter 25 Physics and Simulation.

7.5 Generate and Deform Modifiers

The Generate and Deform Modifier Groups are discussed in Chapters 8 and 9.

8 Editing with Generate Modifiers

8.1 Modifiers - Generate
8.2 Bevel Modifier
8.3 Array Modifier
8.4 Boolean Modifier
8.5 Build Modifier
8.6 Decimate Modifier
8.7 Edge Split Modifier
8.8 Mask Modifier
8.9 Mirror Modifier
8.10 Multiresolution Modifier
8.11 Remesh Modifier
8.12 Screw Modifier
8.13 Skin Modifier
8.14 Solidify Modifier
8.15 Subdivision Surface Modifier
8.16 Triangulation & Weld Modifier
8.17 Wireframe Modifier

Modifiers

As well as the definition offered in Chapter 7, Modifiers may be described as automatic operations that affect an object in a non-destructive way allowing effects to be generated which would otherwise be tedious to do manually.

Modifiers work by changing how an Object is displayed and rendered in the Viewport. The underlying geometry of the mesh is maintained until the

Modifier is Applied. This means that the underlying geometry of the mesh may be edited to suit, before permanently applying the Modifier.

You can add several modifiers to a single Object combining effects. This forms a Modifier Stack.

Modifiers are accessed in the **Properties Editor, Modifiers Properties**.

Figure 8.1

8.1 Modifiers – Generate

The **Generate** group of Modifiers are building tools that change the general appearance of, or automatically add new geometry to an Object.

The Bevel Modifier, used for describing Modifiers in general Chapter 7 – 7.1, falls in this category and is the second Modifier in the alphabetical listing (Chapter 7 Figure 7.2).

For the moment the Bevel Modifier will be used as an example with the Array Modifier and the remaining Modifiers in the Generate Category to follow.

Note: It is not possible to fully describe or explain the function of every button in every Modifier panel. You will discover the detailed use of buttons and functions on a need to know basis as you progress through the book and as you research tutorials on the Internet. To make you aware of how to access Modifier functions the Bevel Modifier will be used as an example.

8.2 The Bevel Modifier

For the general application of the Bevel Modifier refer to Chapter 7 – 7.1. Some of the Functions in the Bevel Modifier are as follows:

1 How the Bevel affects the Object.
Figure 8.3 below.

2 Width Type Options.

3 Width Amount.

4 Bevel Segments.

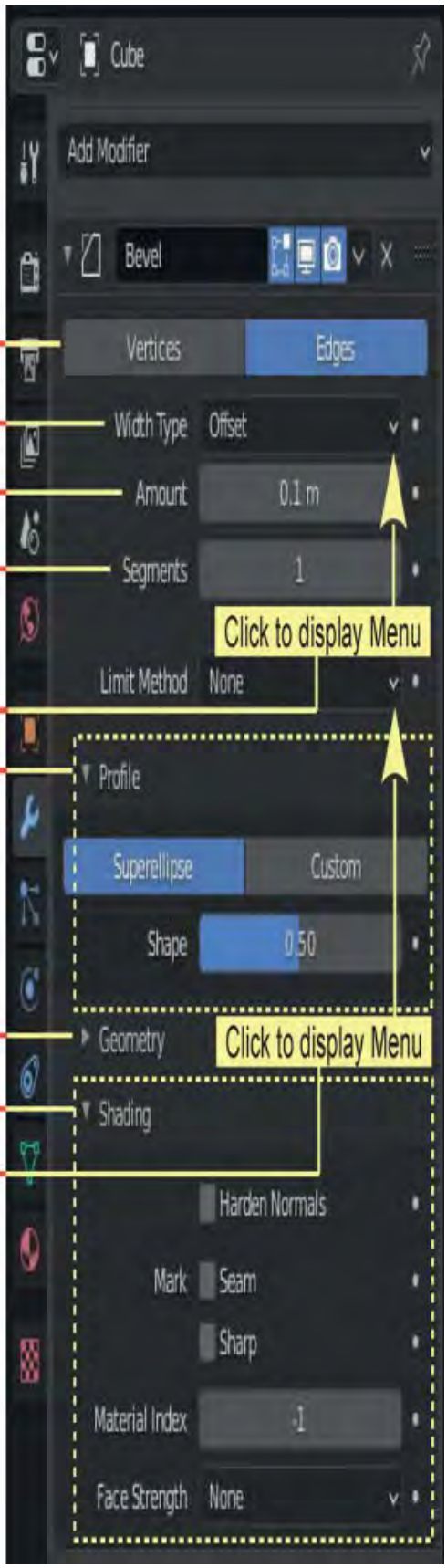


Figure 8.2

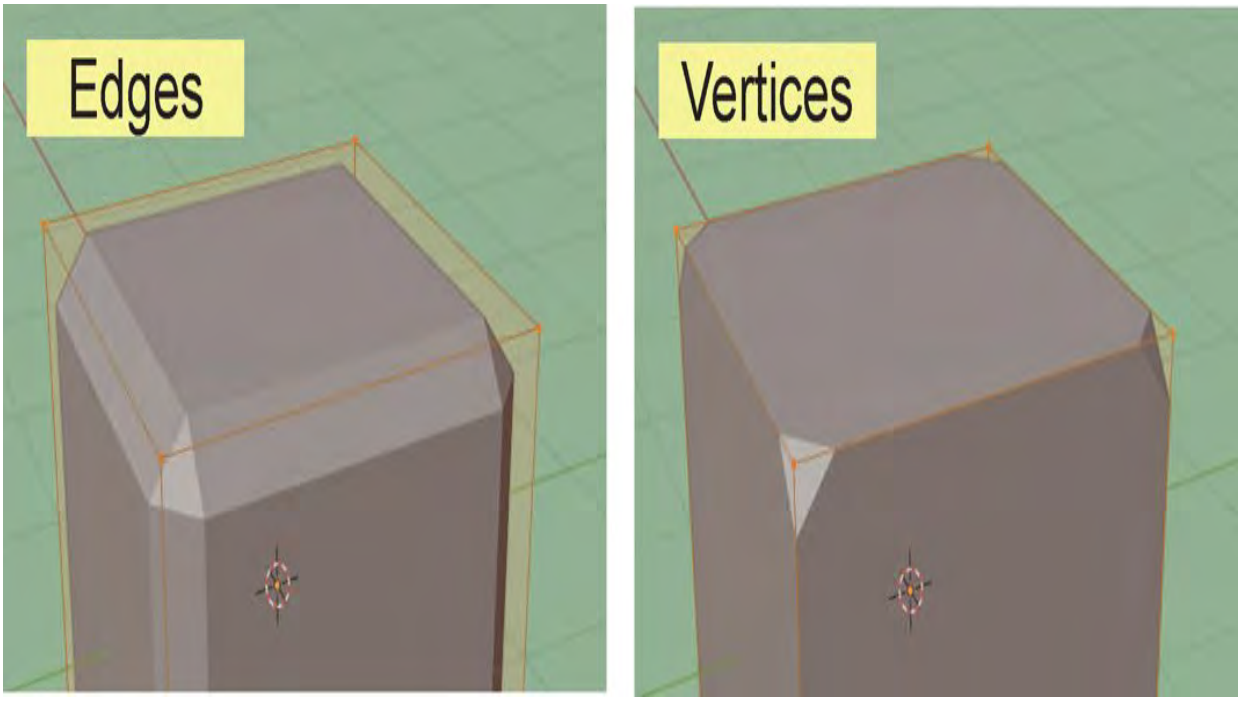


Figure 8.3

1. How the Bevel affects the Object. Figure 8.3 below.
2. Width Type Options.
3. Width Amount.
4. Bevel Segments.

Vertices/Edges

> **Note:** If you use the Bevel Tool in Edit Mode then add a bevel Modifier in Object Mode you will be bevelling the bevel.

8.3 Array Modifier

The **Array Modifier** creates copies of an Object, placing the copies in an array with each copy offset from the original. Figure 8.5 shows a **Monkey**

Object in **Front Orthographic** view duplicated using an Array Modifier. To add the modifier select the Monkey in the 3D Viewport Editor then in the **Properties Editor, Modifier Properties** click on **Add Modifier** and select **Array** from the menu.

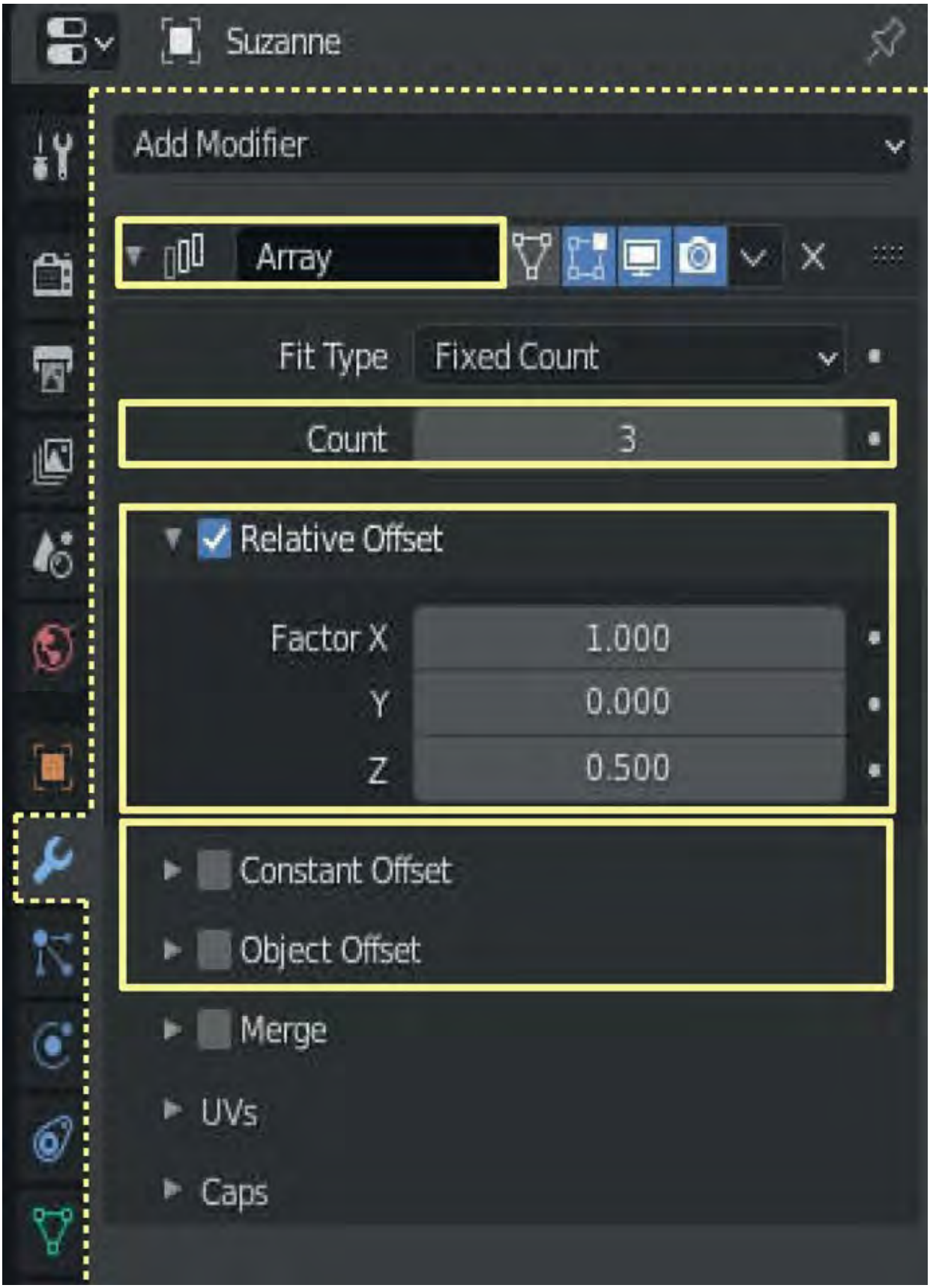

Suzanne
Add Modifier
Array
Fit Type
Fixed Count
Count
3
Relative Offset
Factor X
1.000
Y
0.000
Z
0.500
Constant Offset
Object Offset
Merge
UVs
Caps

Figure 8.4

To produce the arrangement shown in Figure 8.5 enter the **Relative Offset** values shown in Figure 8.4. The **Count: 3** value tells Blender to produce three Monkeys in the array (the original plus two).

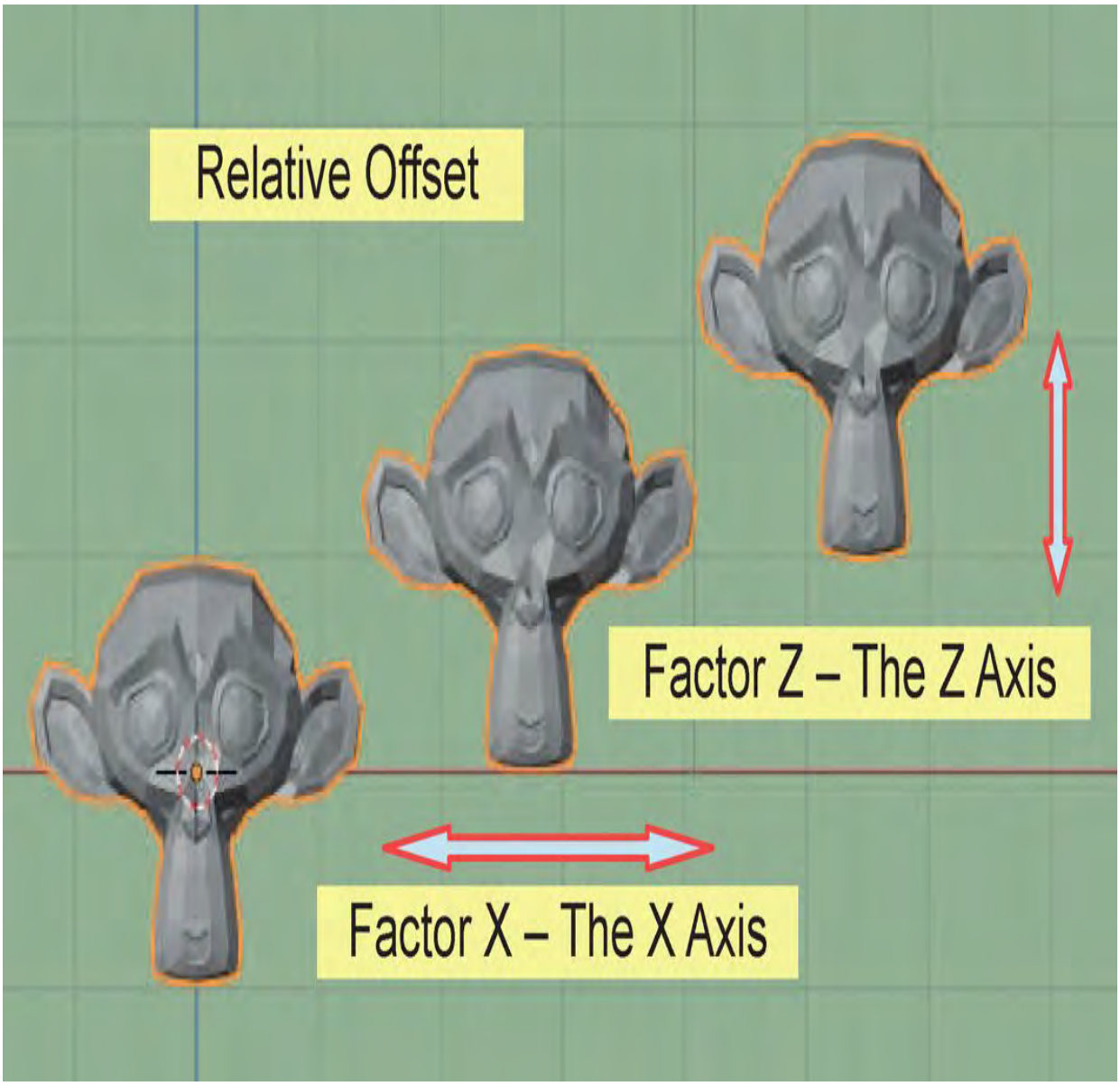


Figure 8.5

The Offset is calculated using the **Object's Bounds**. Every Object has a **Bounding Volume** which encapsulates its shape. You may view the Bound Volume by checking (tick) **Bounds** in the **Properties Editor, Object Properties, Viewport Display Tab**.

The difference between relative and constant offset is shown in Figure 8.6. Constant offset one (1) means offset one Blender unit irrespective of the Object's size. The Monkey's center is offset one Blender unit which overlaps the display. Relative offset uses the **Bound** size (overall size) of the Object.

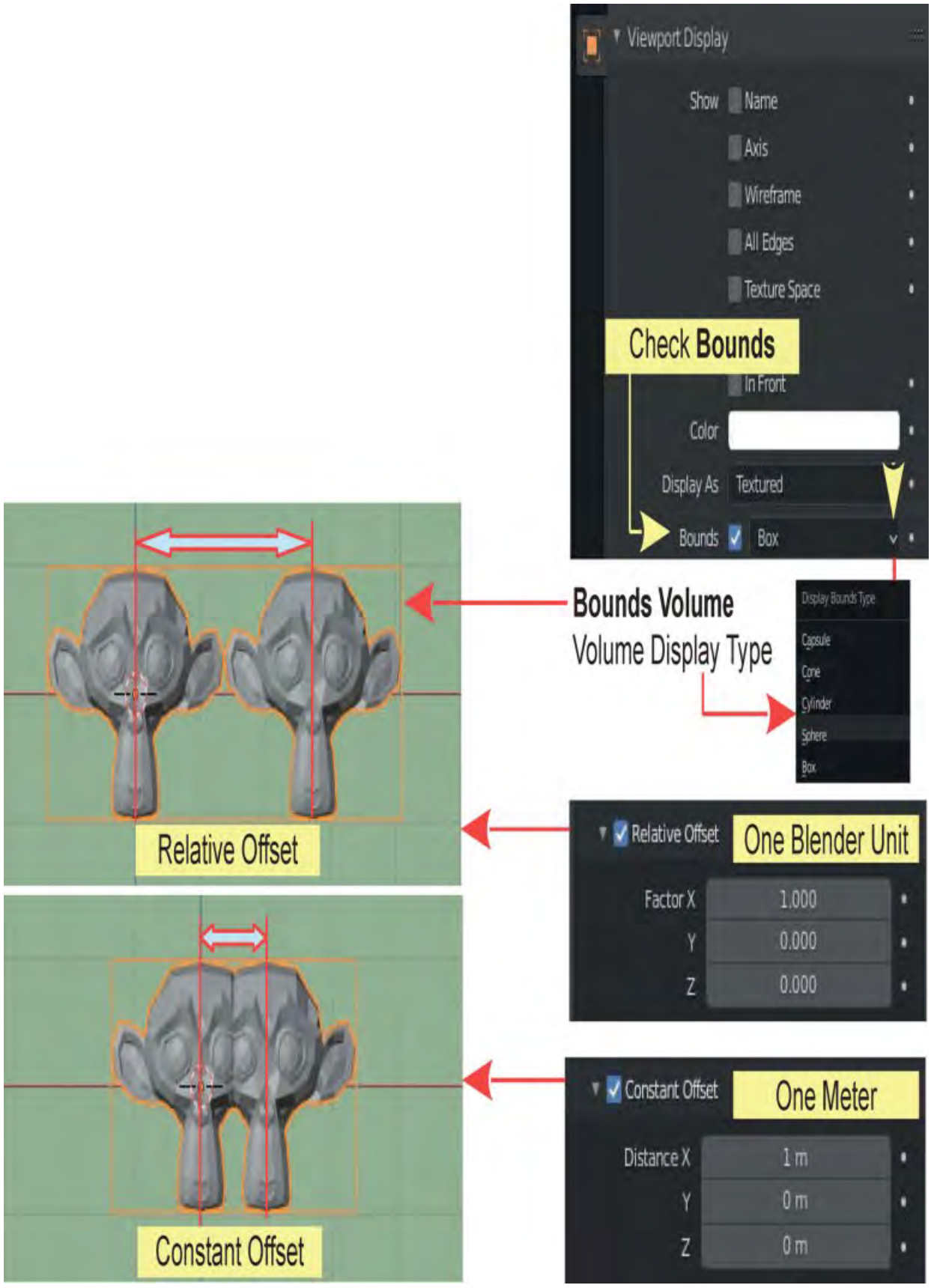

Viewport Display
Show
Name
Axis
Wireframe
All Edges
Texture Space
Check Bounds
In Front
Color
Display As
Textured
Bounds
Box
Bounds Volume
Volume Display Type
Display Bounds Type
Capsule
Cone
Cylinder
Sphere
Box
Relative Offset
Relative Offset
One Blender Unit
Factor X
1.000
Y
0.000
Z
0.000
Constant Offset
Constant Offset
One Meter
Distance X
1 m
Y
0 m
Z
0 m

Figure 8.6

Object Offset: Uses the relative displacement of one Object to influence the displacement of another.

To use **Object Offset** position the **Control Object** (UV Sphere) in the 3D Viewport Editor (you can Translate, Rotate and Scale after adding the modifier if you wish). By default **Relative Offset** is checked (ticked) in the **Properties Editor, Modifier Properties**. Change to **Constant Offset** and check **Object Offset**. Click in the bar below Object Offset and select your **Offset Object** (the Sphere) from the menu that displays (Figures 8.7, 8.8).

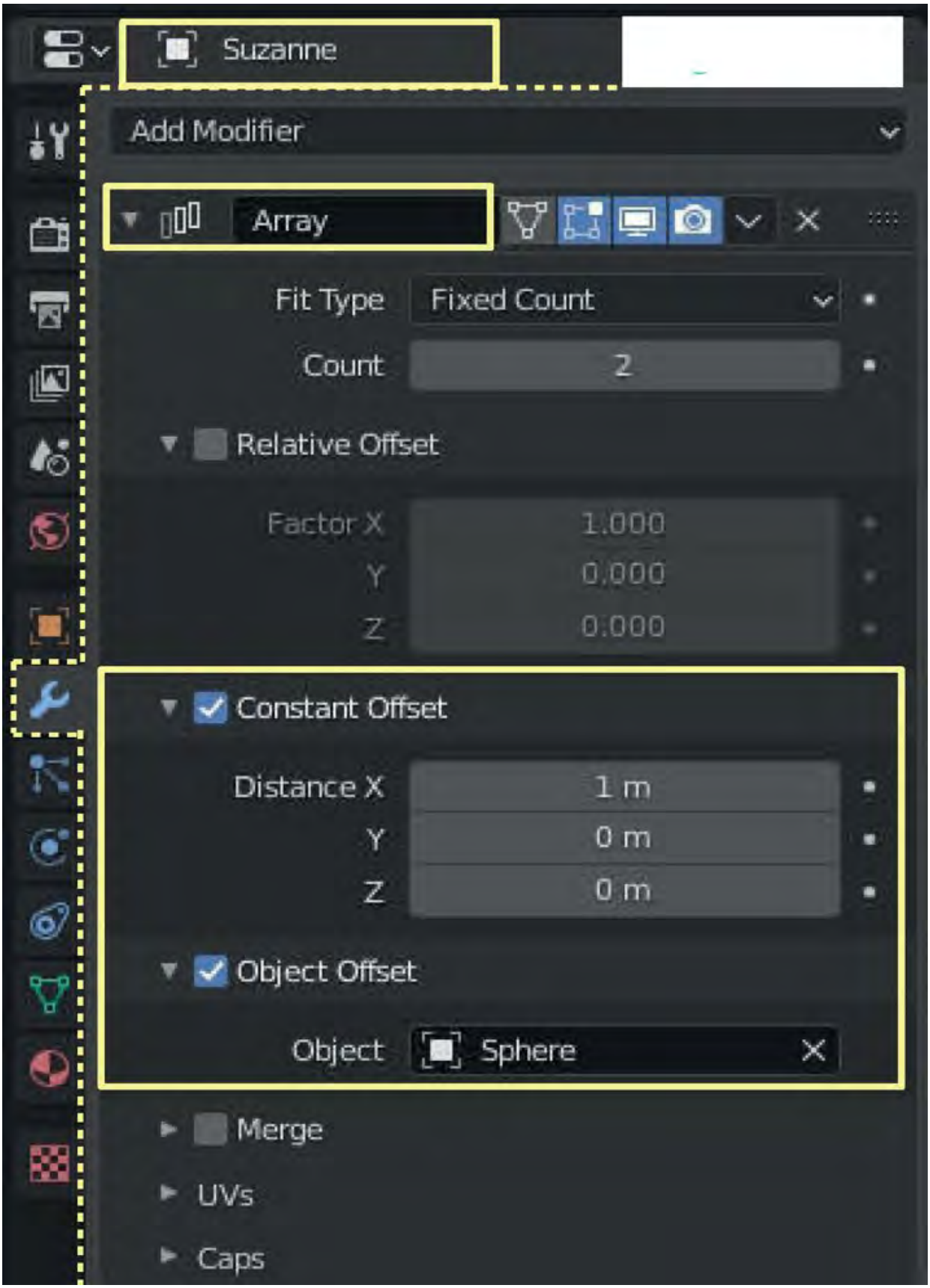

Suzanne
Add Modifier
Array
Fit Type
Fixed Count
Count
2
Relative Offset
Factor X
1.000
Y
0.000
Z
0.000
Constant Offset
Distance X
1 m
Y
0 m
Z
0 m
Object Offset
Object
Sphere
Merge
UVs
Caps

Figure 8.7

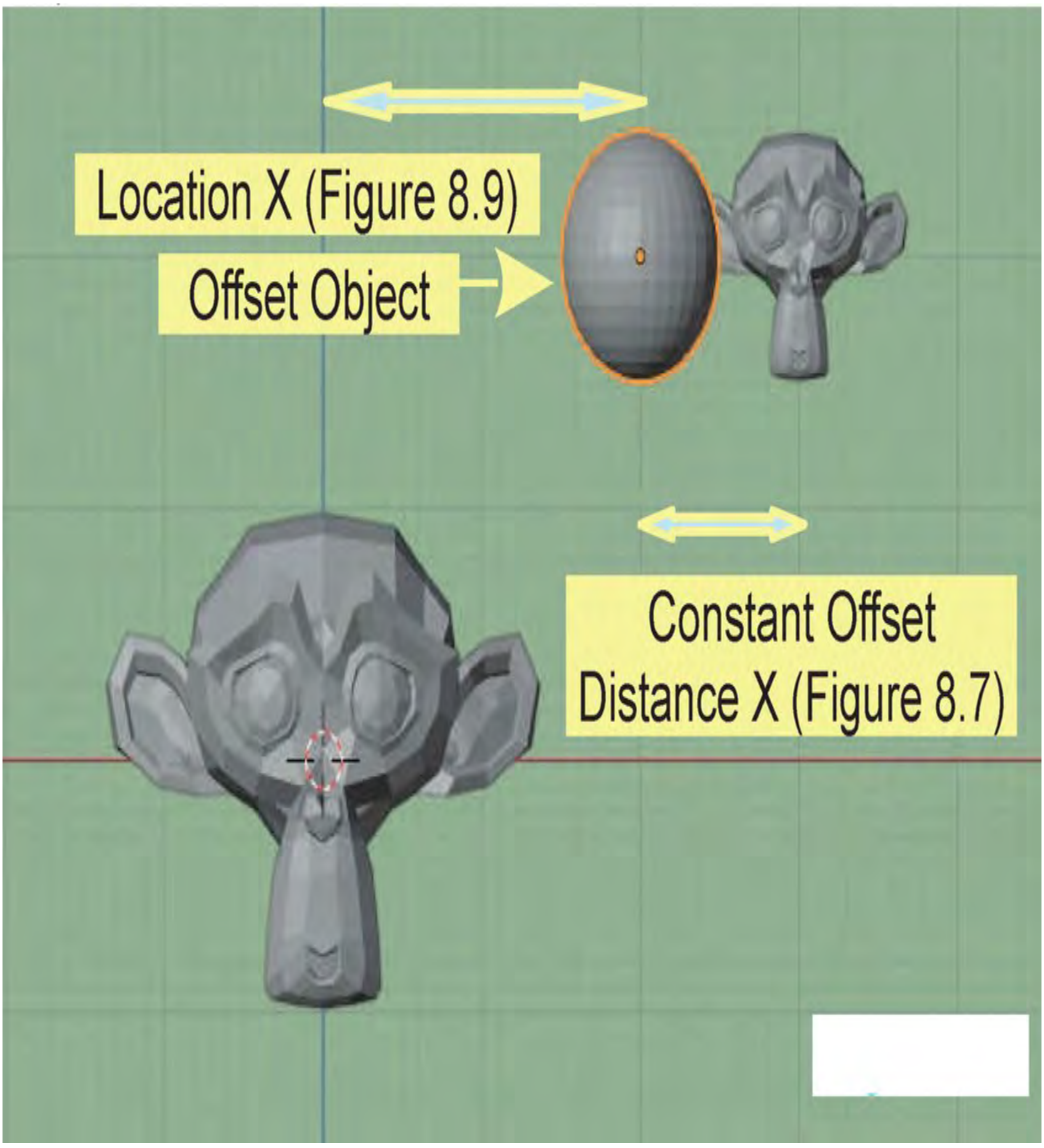


Figure 8.8

In Figure 8.8 **Suzanne** (Monkey) has an Array Modifier added (Figure 8.7) with Constant Offset and Object Offset checked. The Viewport shows Front

Orthographic View. A UV Sphere has been placed in the Scene and positioned plus two (2) meters on the X Axis and plus two (2) meters on the Z Axis (Figure 8.9). The Sphere is Scaled down 0.500.

With Count 2 set in the Array Modifier for Suzanne and Object Offset selected with Object Offset, Object: Sphere, a duplicate is generated half size and positioned plus one meter offset on the X Axis away from the Sphere.

Sphere
Sphere
Transform
Location X
2 m
Y
-0 m
Z
2 m
Rotation X
0°
Y
0°
Z
0°
Mode
XYZ Euler
Scale X
0.500
Y
0.500
Z
0.500
Delta Transform
Visibility
Selectable
Show in
Viewports
Renders

The position of the Object Offset control Object (Sphere) is set in the **Object Properties** for the Sphere (Figure 8.9)

Note: When a Control Object is used in the Array Modifier it will Render even though you cancel the display in the 3D Viewport Editor by clicking the Eye Icon adjacent to Sphere in the Outliner Editor. To prevent the Control Object from Rendering, select it tn the 3D Viewport then in the Properties Editor, Object Properties, Visibility Tab, uncheck **Renders**.

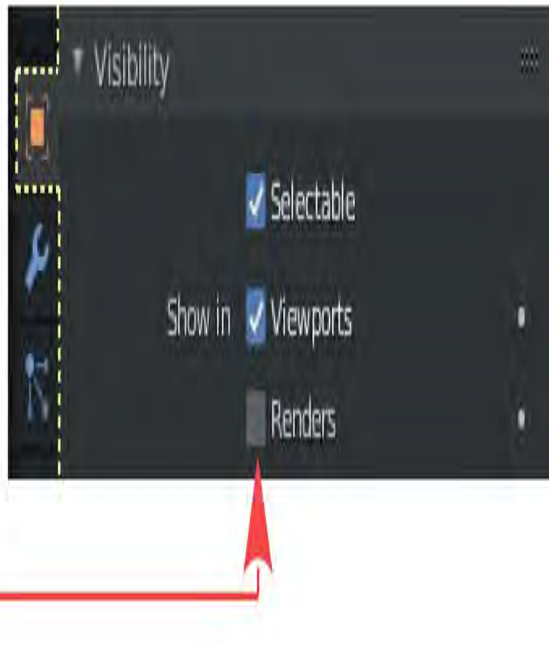


Figure 8.10

Increasing the Count Value in the Array Modifier panel duplicate Objects. As demonstrated, when using Object Offset, the duplicated Object is positioned and Scaled accordingly. Increasing the Count Value in the Modifier will replicate the original Object, repositioning and Scaling exponentially. Any modification made to the Control Object will be reflected in the duplications. (see Figure 8.11).

Figure 8.11 shows the Array with the Count Value in the Modifier increased to 4 and with the UV Sphere Control Object Scaled and Rotated.

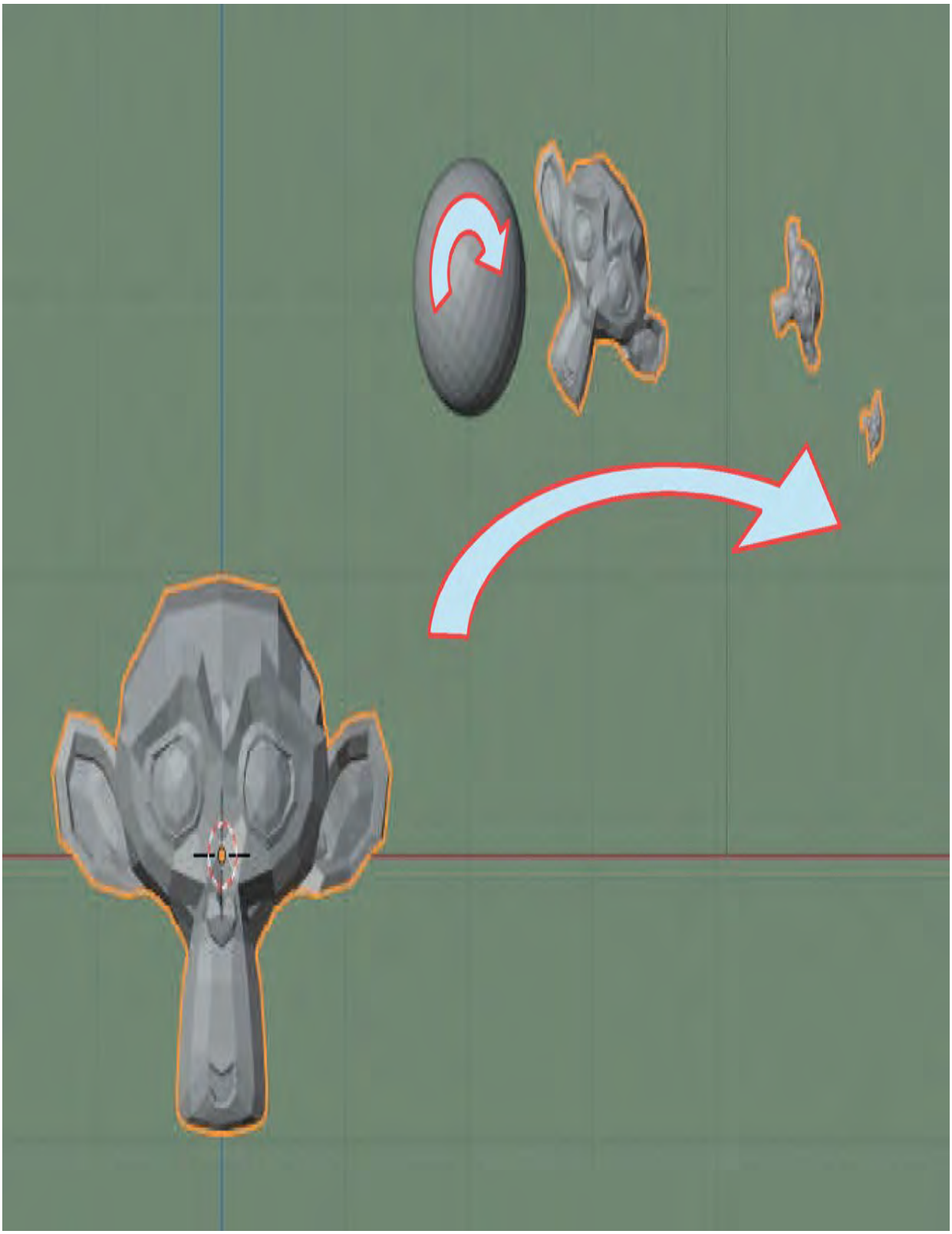


Figure 8.11

8.4 Boolean Modifier

Boolean Modifiers are used to create shapes by adding or subtracting one Object from another. In the Modifier panel there are three options: **Difference**, **Intersection**, and **Union.**

To demonstrate the different operations position a UV Sphere Object with the default Cube as shown in Figure 8.12. Scale the sphere down to fit inside the top face of the cube.

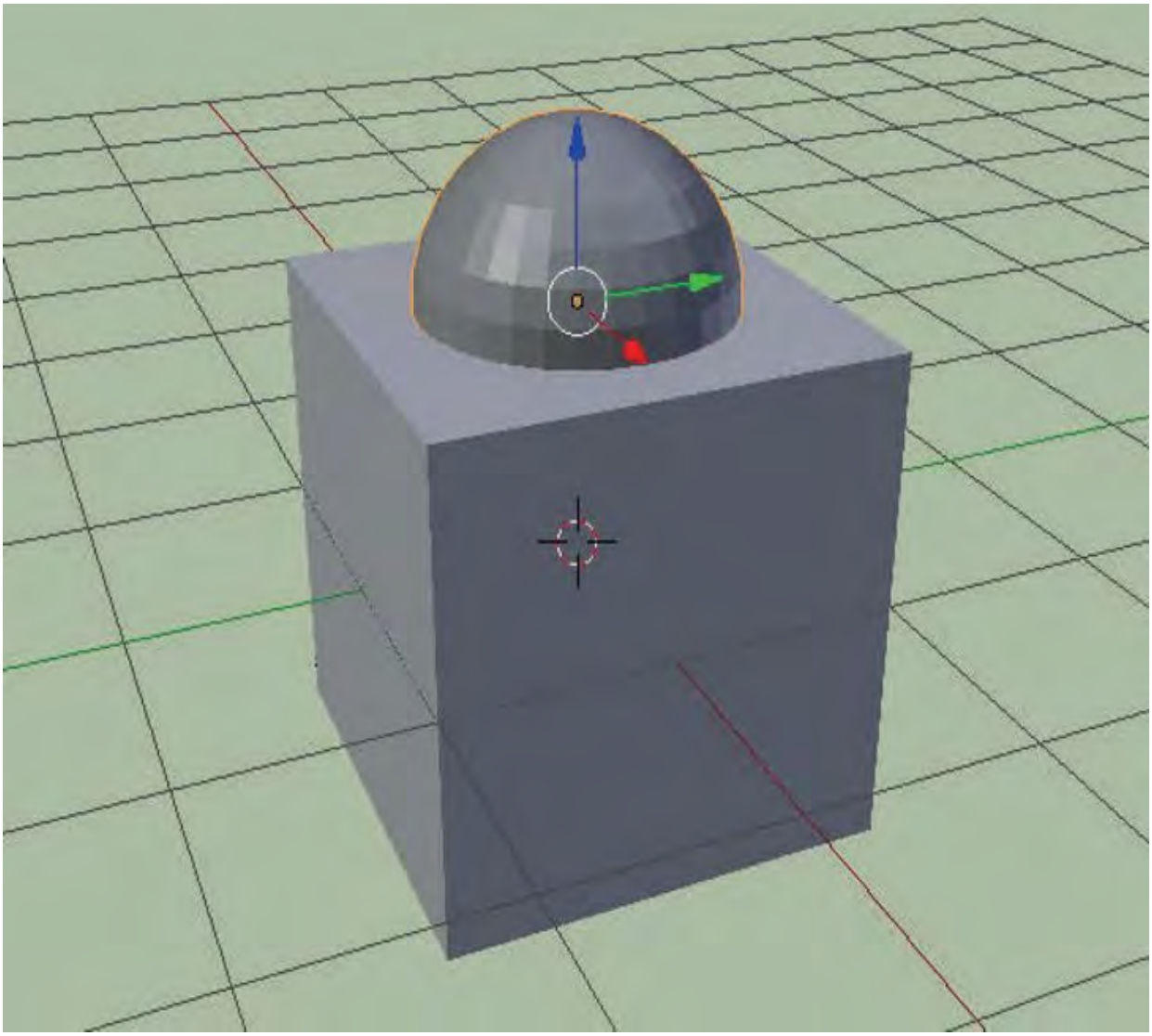


Figure 8.12

The arrangement of cube and sphere will be used for all three Boolean operations.

The procedure for a Boolean Operation is as follows; Select the Cube. Add the Modifier. Select the Operation Type. Enter the Boolean Object. Click Apply (Figure 8.13)

Boolean Difference

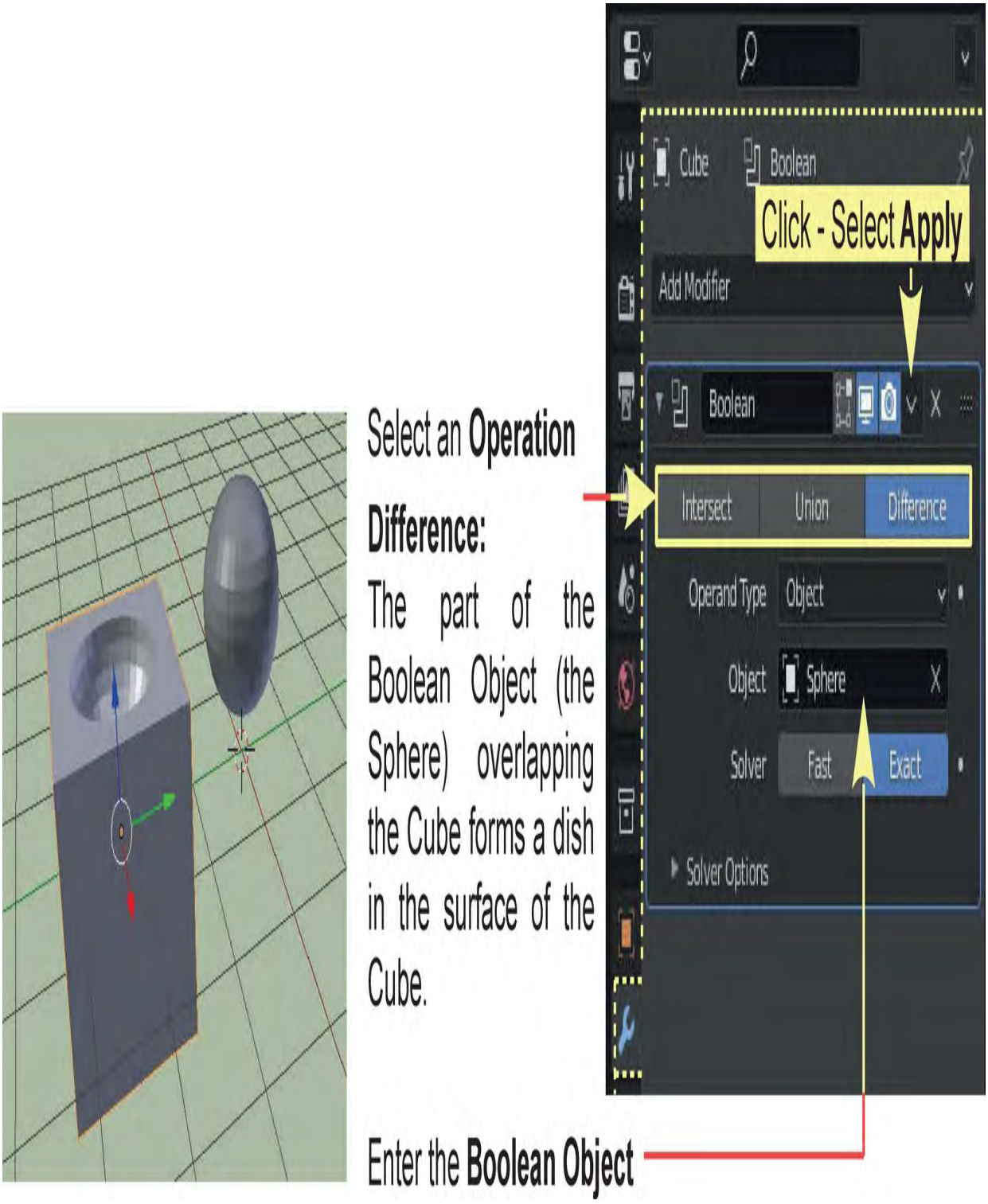


Figure 8.13

Boolean Intersect

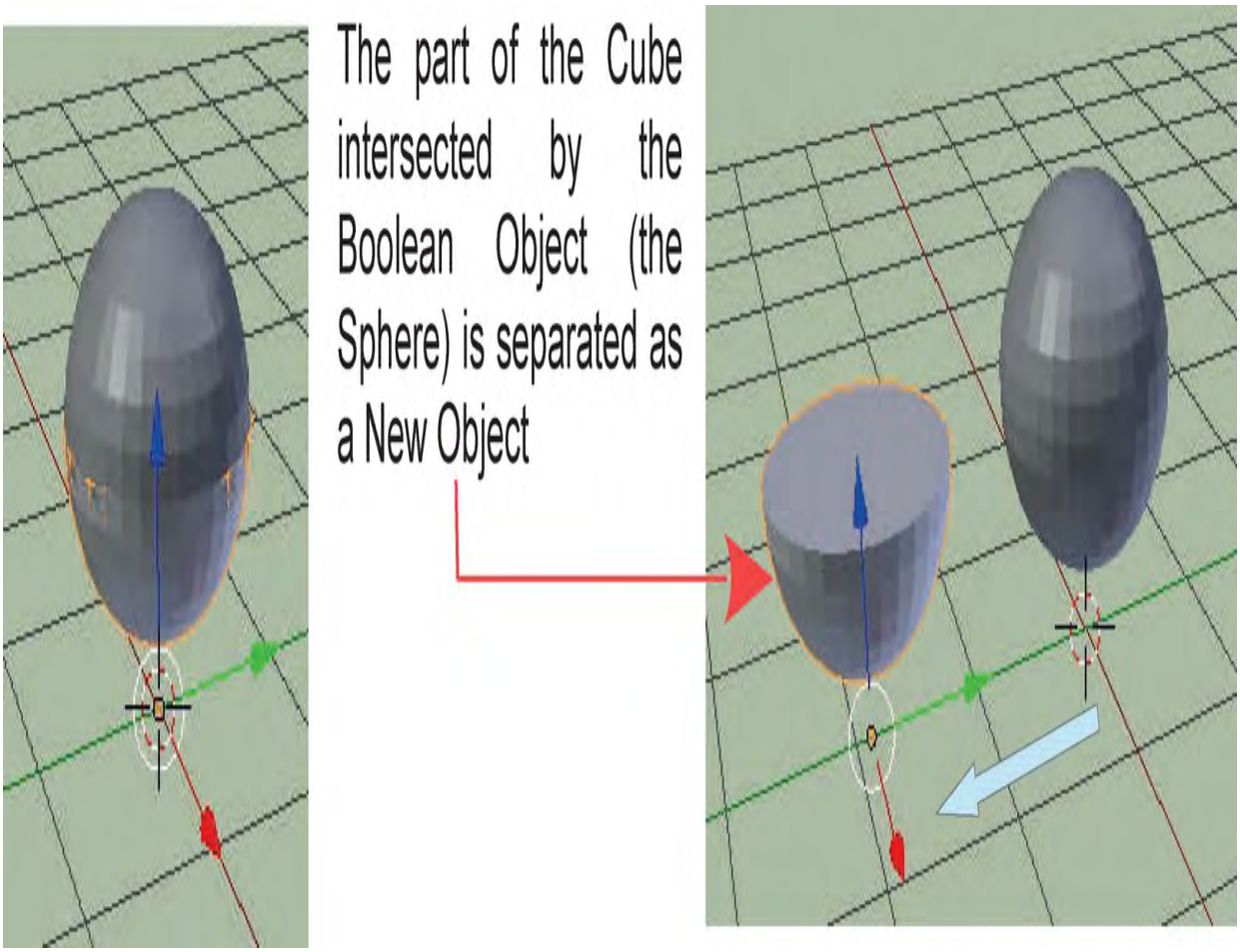


Figure 8.14

Boolean Union

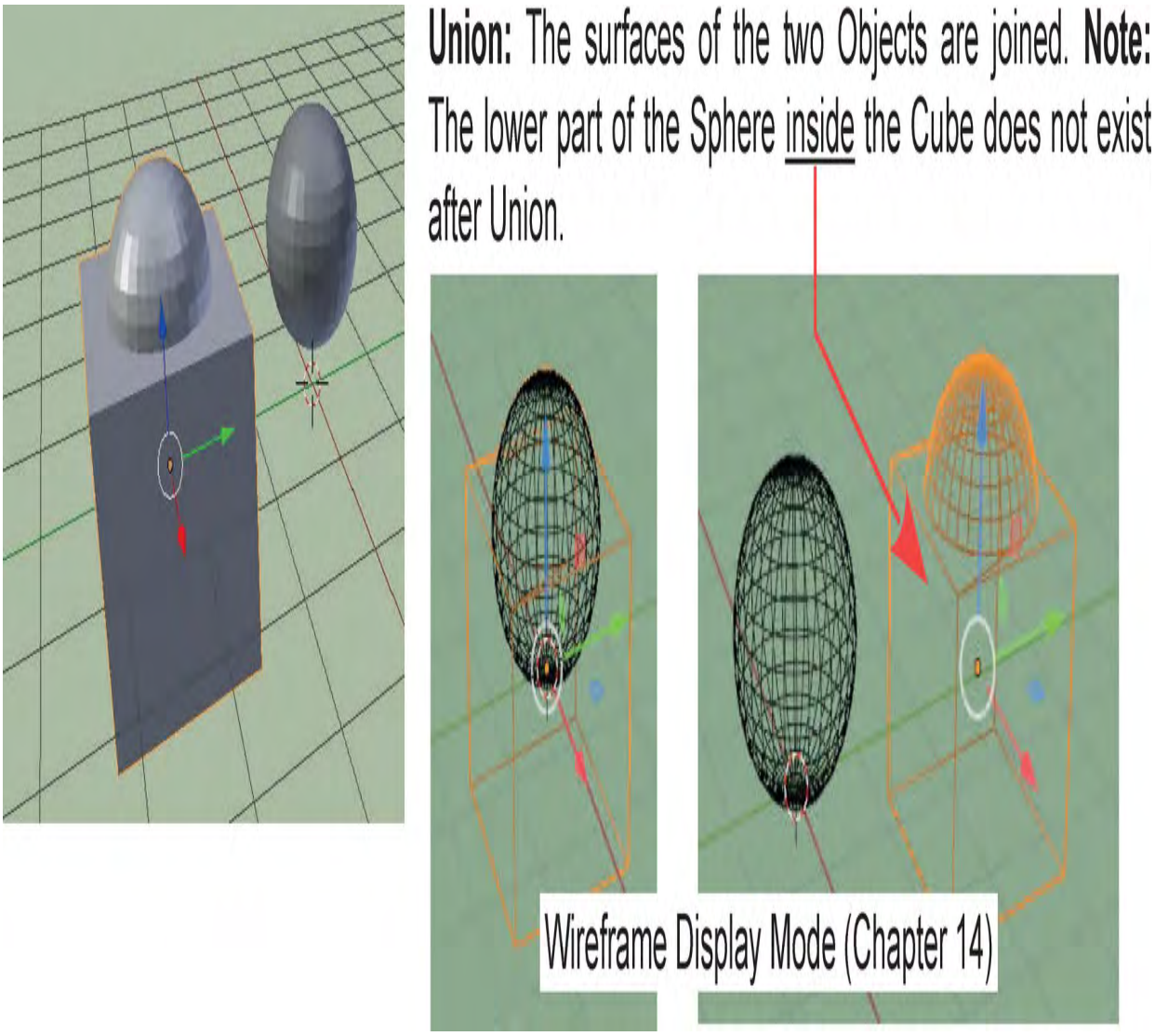


Figure 8.15

8.5 Build Modifier

The **Build Modifier** creates the effect of something building linearly over a period of time. Any Object can have a build modifier, but to see a nice effect, a high vertex count is required.

In the 3D View Editor, Scale the default Cube on the Y axis, Tab to **Edit mode** and Subdivide the surface (Chapter 5 – 5.5 Number of Cuts 10).

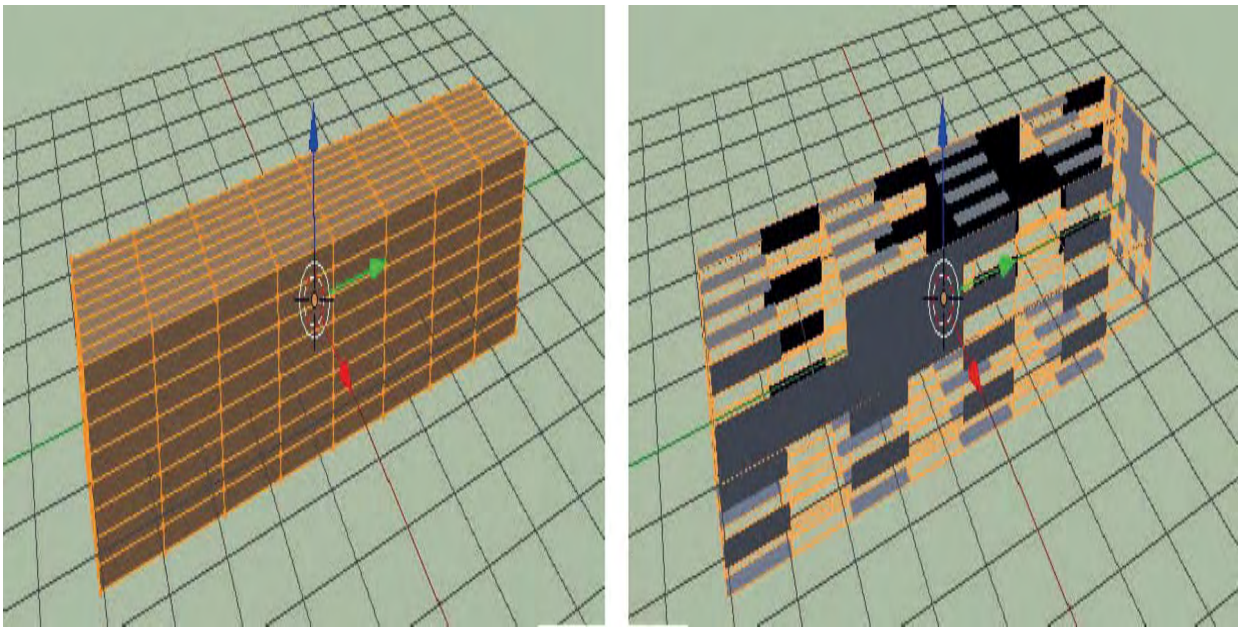


Figure 8.16

Tab back to **Object mode** and add a **Build Modifier** (Figure 8.19). **The Object disappears from view**.

When adding a Build Modifier you have, in effect, created an Animation of the Object building over time. In the **Timeline Editor** at the bottom of the Screen you will see a blue rectangle with a blue line at Frame 1. The graduations; Frame 1 to Frame 250 indicate that the Animation Length is 250 Frames. The significance of this is explained in Chapter 20.

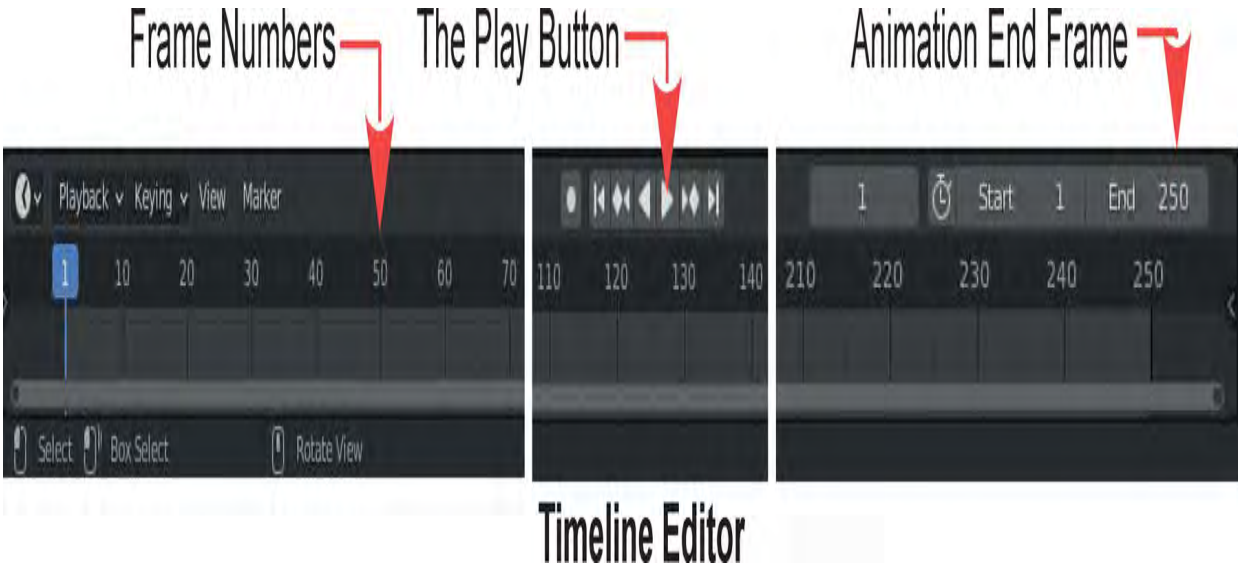


Figure 8.17

You may click, hold and drag the blue rectangle to the right to see the Object in the 3D Viewport Editor reconstruct in sections.

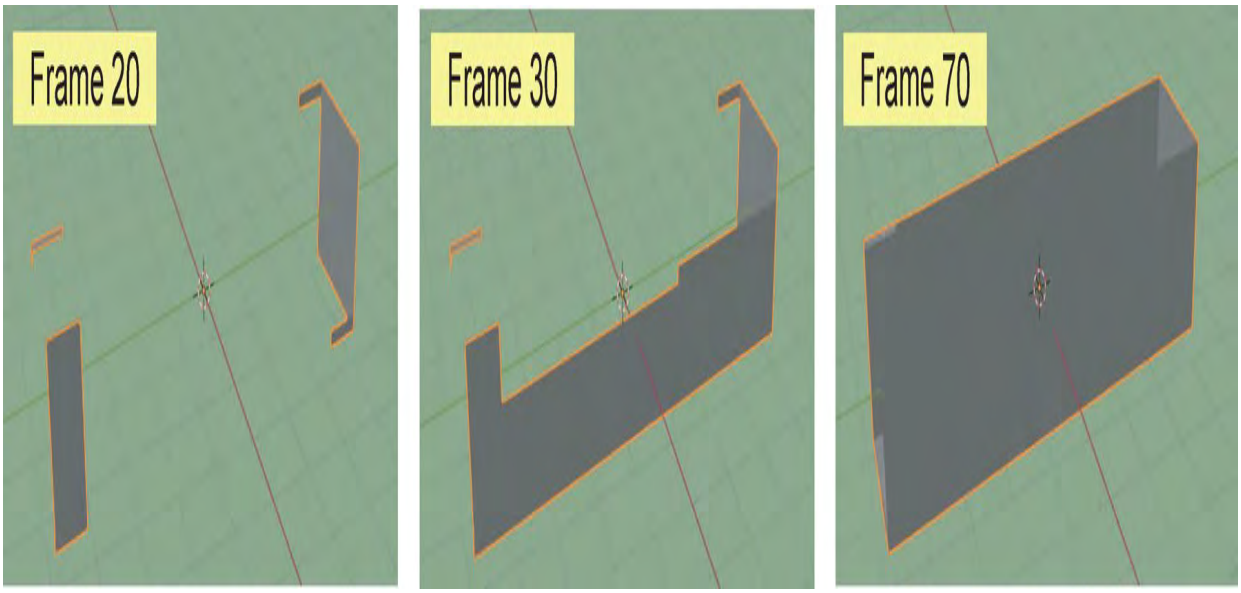


Figure 8.18

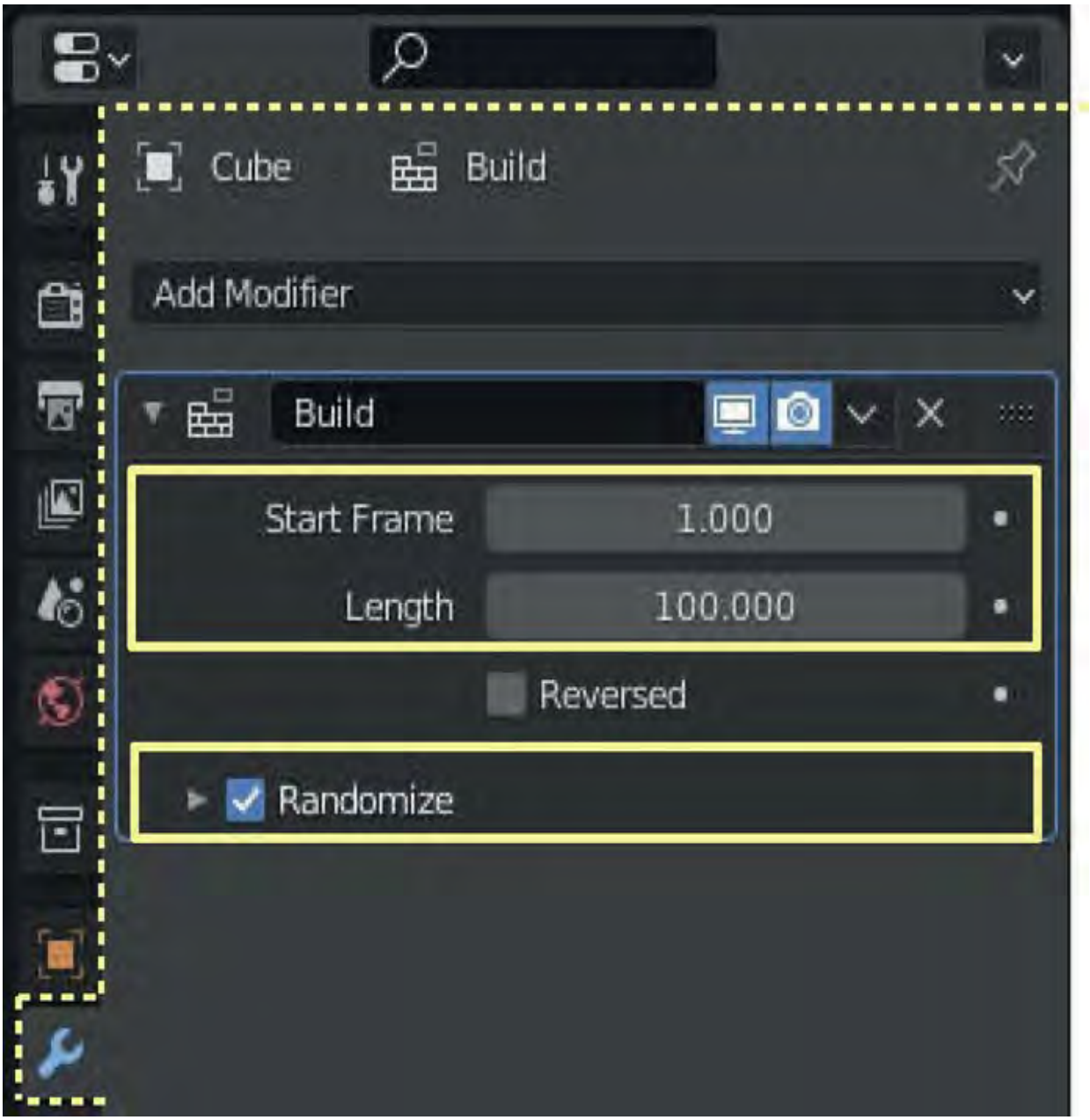


Figure 8.19

By pressing the **Play Button** in the Timeline Editor you play the Animation showing reconstruction.

Note: The Object is fully reconstructed at Frame 100 due to the **Length** setting in the Properties Editor, Build Modifier (Figure 8.19).

For a more dramatic reconstruction (Figure 8.20), check **Randomise** in the Modifier.

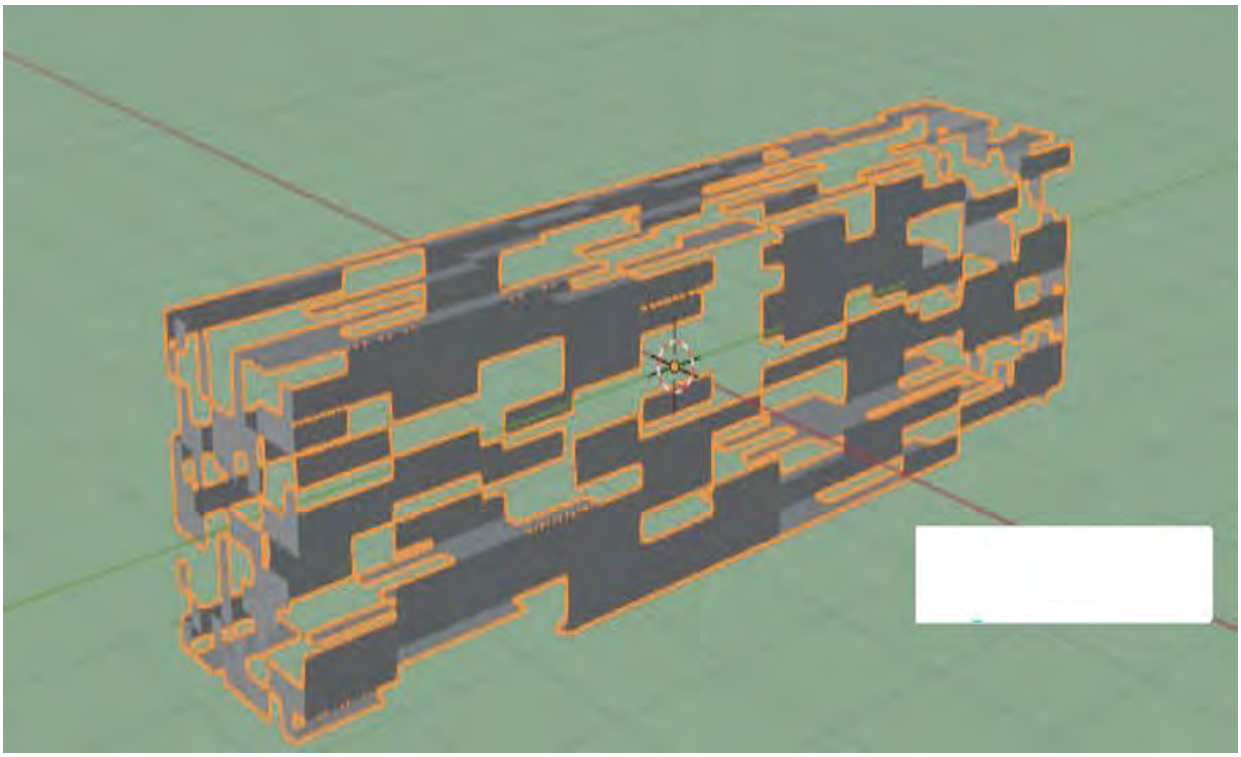


Figure 8.20

8.6 Decimate Modifier

When a mesh Object has been created using complex modeling, you may well have an Object with many vertices and , therefore, a high Vertex Face count. Blender uses the Vertex/Face count to calculate such things as shading effects. This should not be confused with the Vertices and mesh Faces in the actual construction of a model. The Vertex/Face count is, in effect, triangulation within mesh Faces.

Using the Decimate Modifier is a quick and easy way of reducing the Vertex/Face count.

To demonstrate Decimation start with the default blender Scene, delete the Cube Object and add a **Monkey**. The Monkey is a reasonably complex shape consisting of numerous Faces and Vertices.

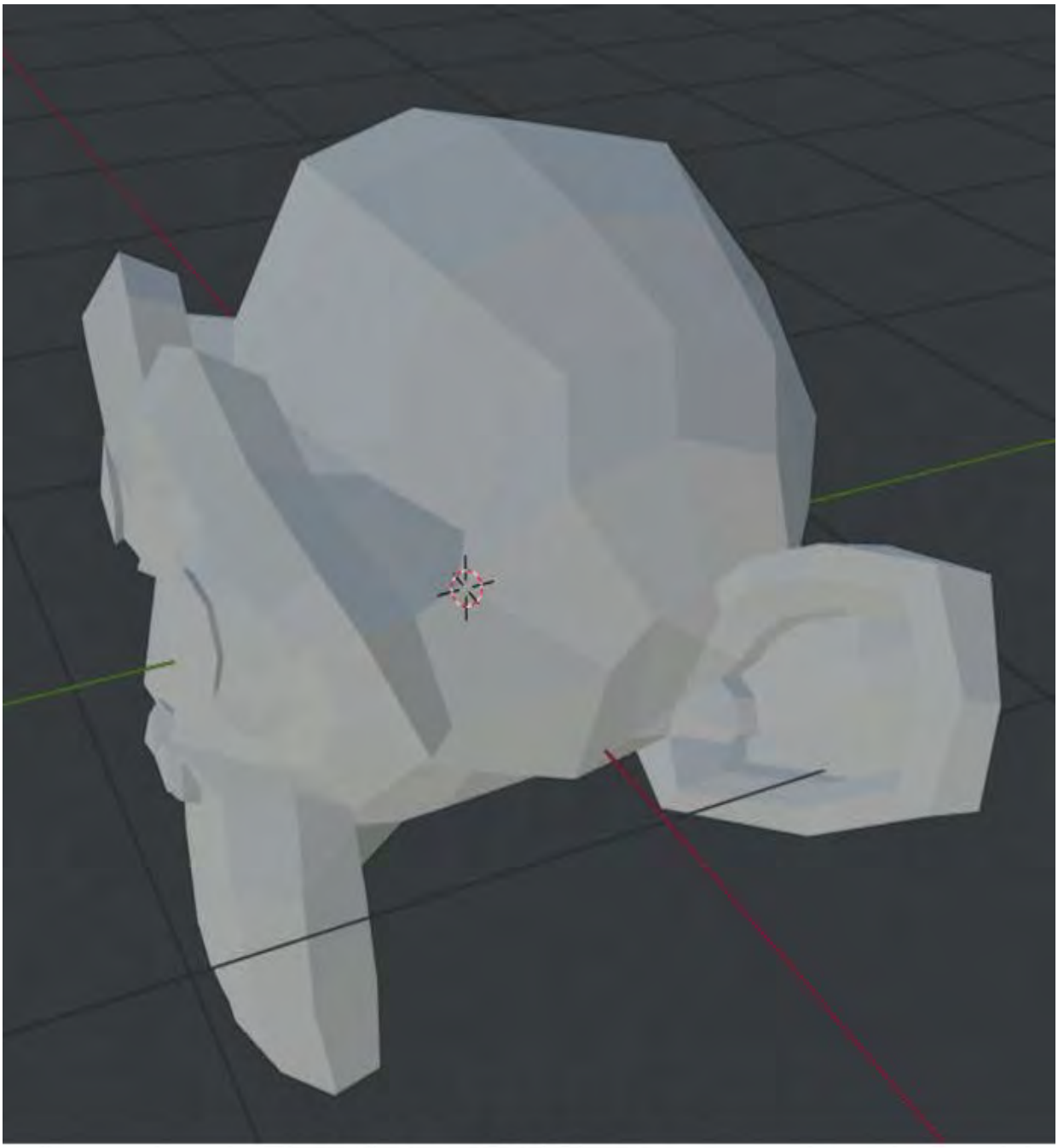


Figure 8.21

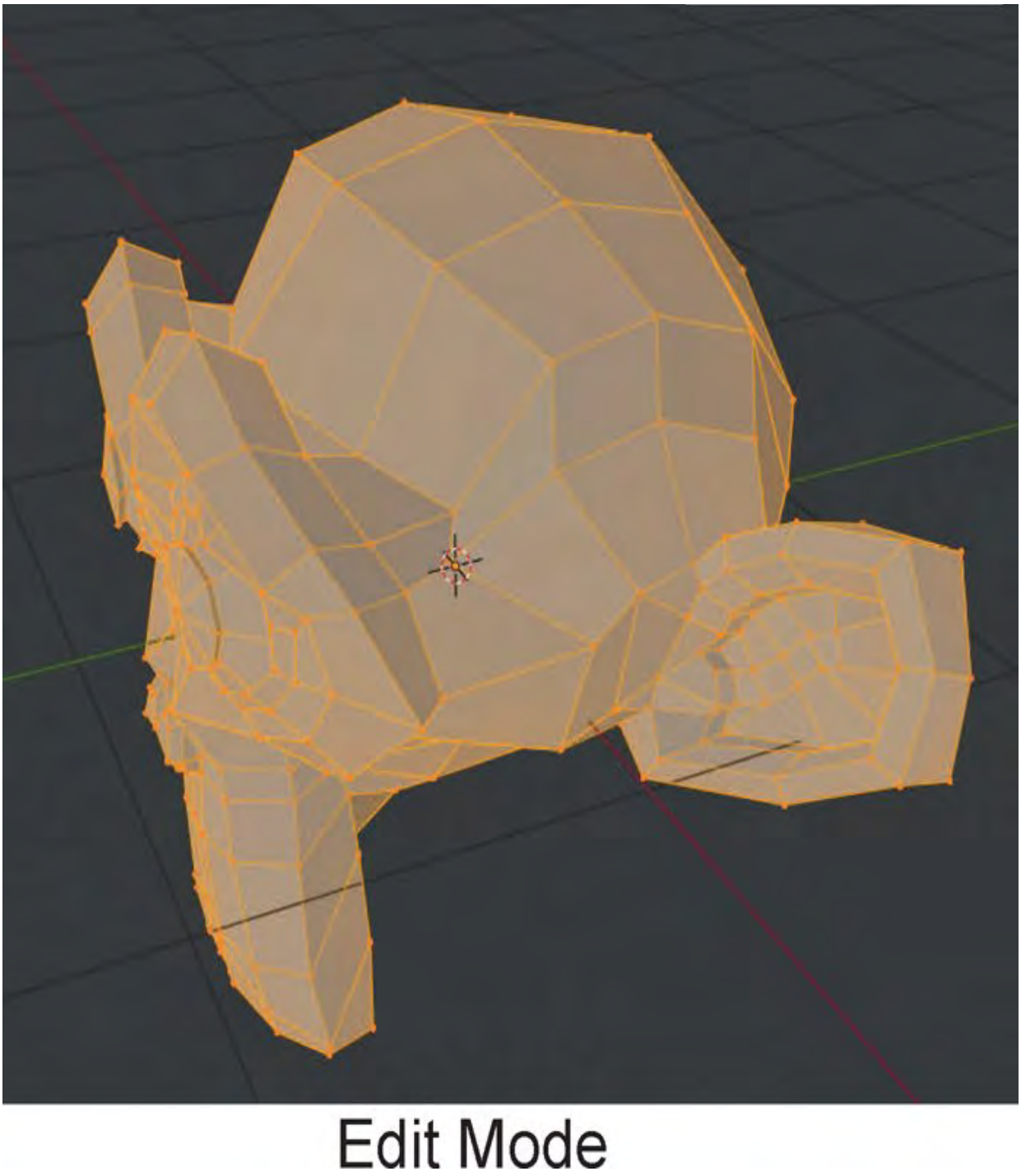


Figure 8.22

In Object Mode the shape of the Monkey is representative of the Vertices you see in Edit Mode.

Applying a Decimate Modifier and reducing the Ratio value to approximately 0.25 (Figure 8.24) significantly alters the shape in the 3D View Editor (Figure 8.23).

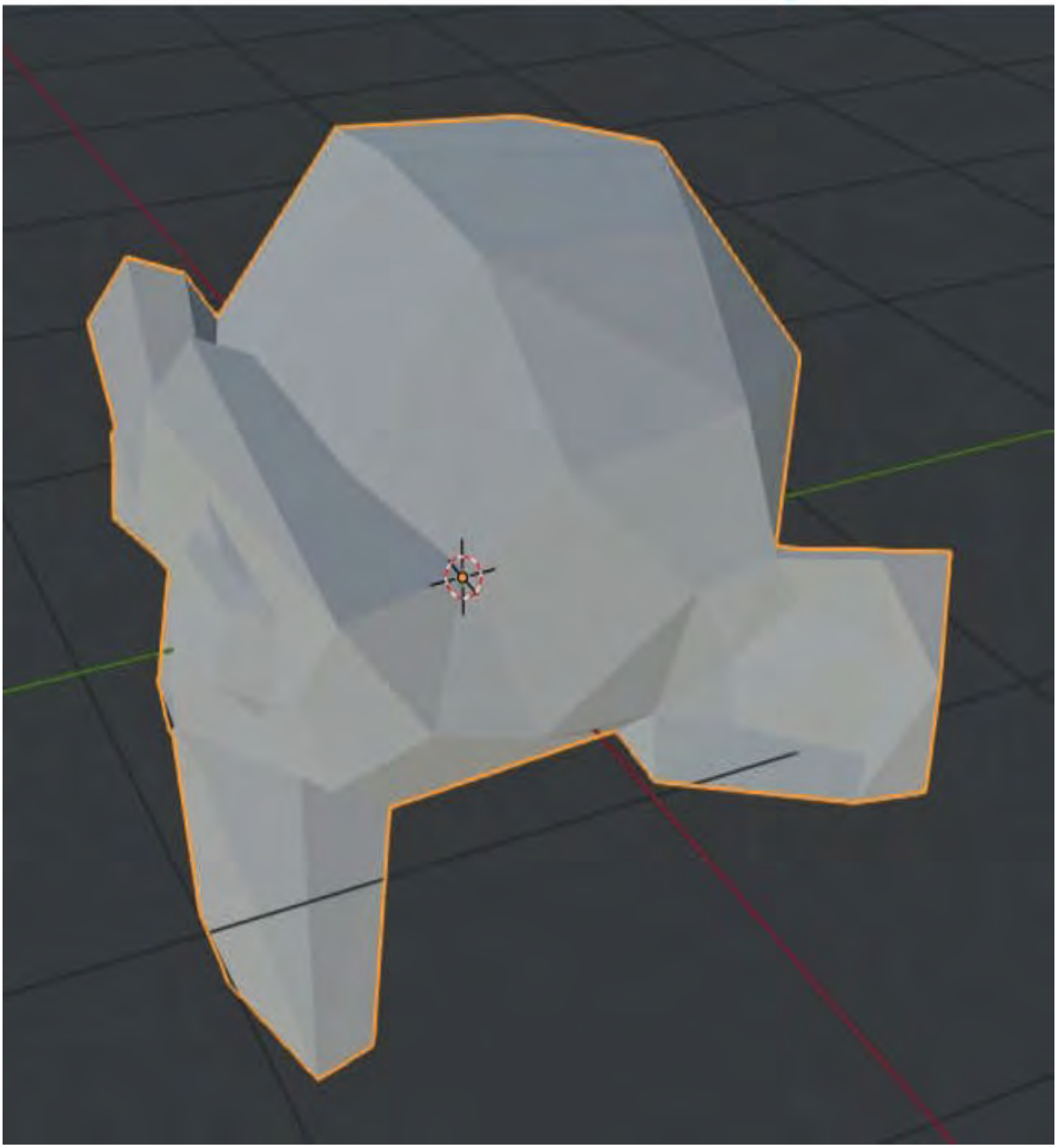


Figure 8.23

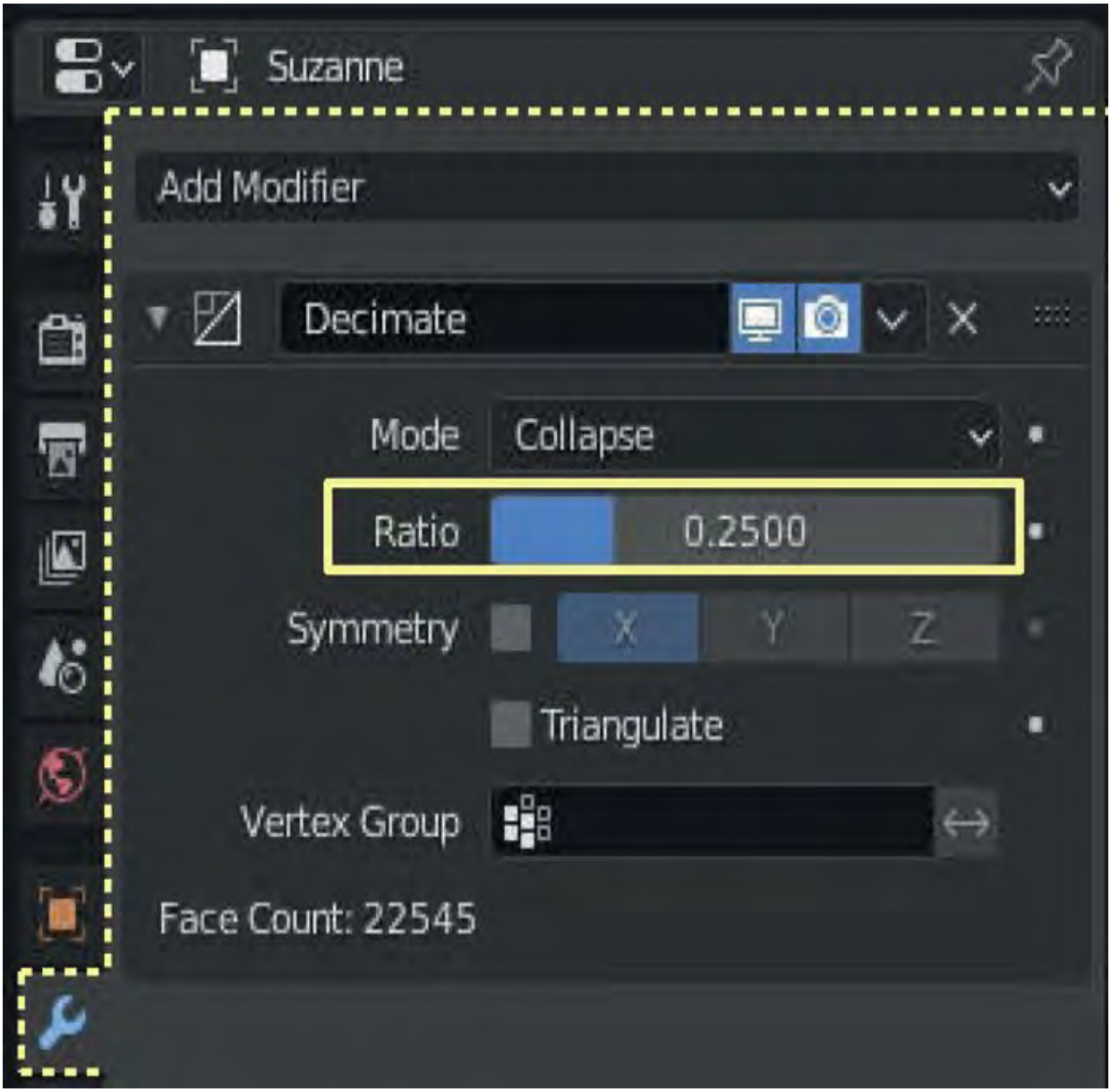


Figure 8.24

Starting over with the original Monkey Object and Subdividing with Count 10 in Edit Mode produces many Vertices (Figure 8.25). With this increased number of Vertices Applying a Decimate Modifier and reducing the Ratio value to 0.25 has no appreciable effect on the display in the 3D Editor (Figure 8.26) or in a rendered View.

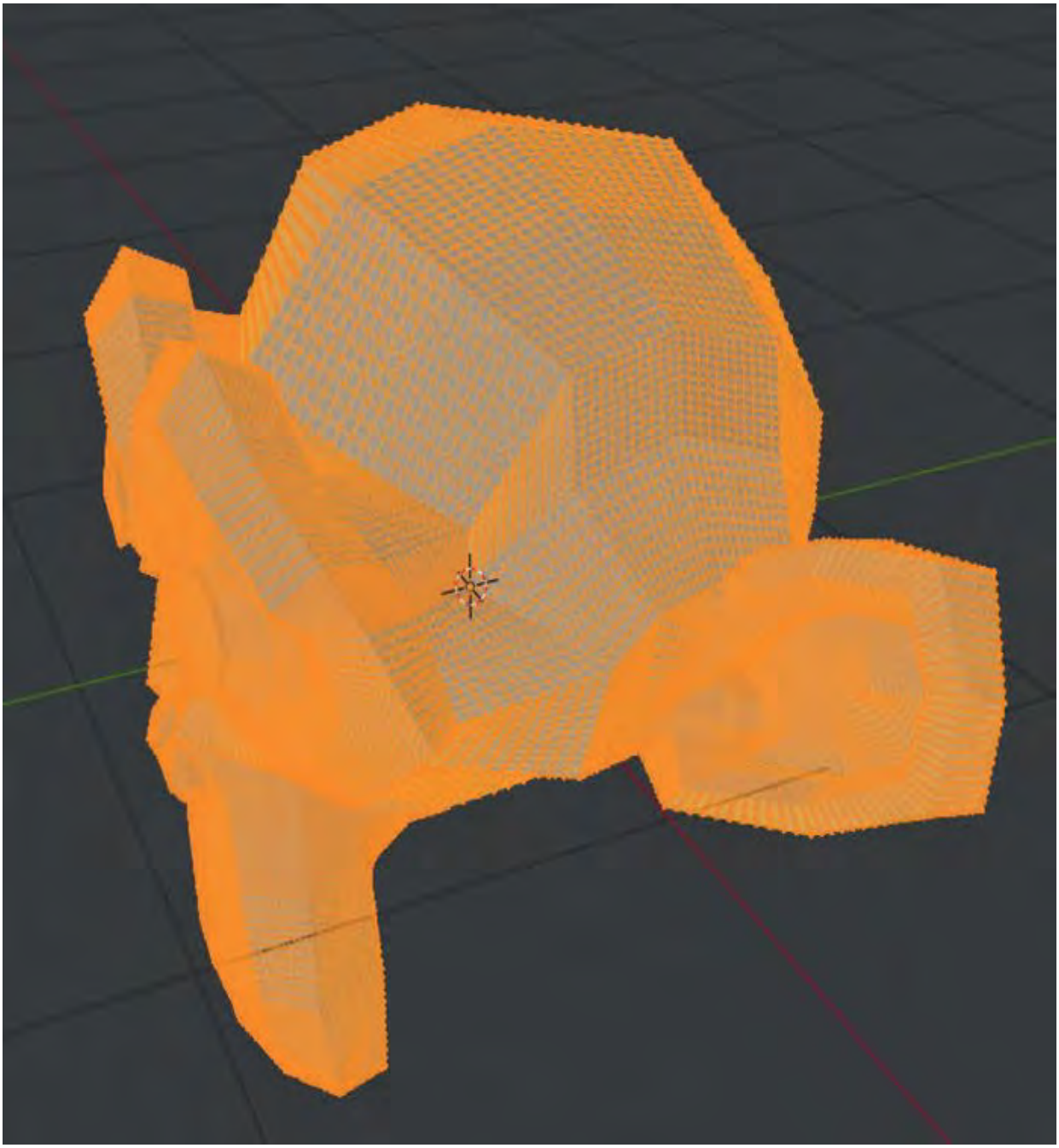


Figure 8.25

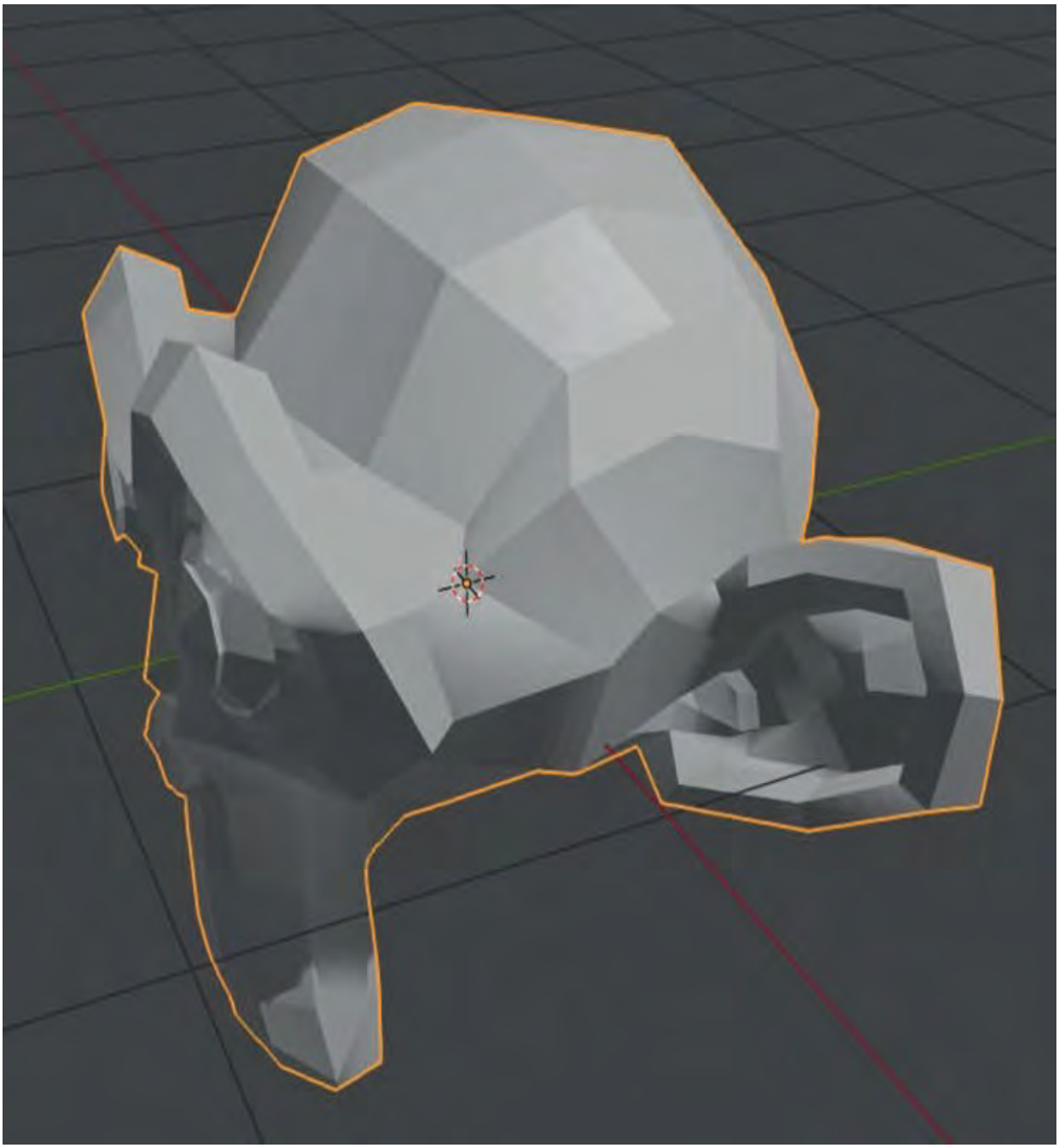


Figure 8.26

Reducing the Ratio Value to 0.255 in the Decimate Modifier when there are many Vertices reduces the Face Count for generating Shading Effects.

8.7 Edge Split Modifier

Although named **Edge Split** the **Edge Split Modifier** allows you to split an Object apart by selecting **Vertices**, **Edges**, or **Faces**.

To demonstrate: Select the Cube Object in the default 3D Viewport Editor in Object Mode and **Add and Apply** an **Edge Split Modifier** (Figure 8.27). The Modifier panel in the Properties Editor disappears when the Modifier is Applied. Tab to Edit Mode after Applying the Modifier.

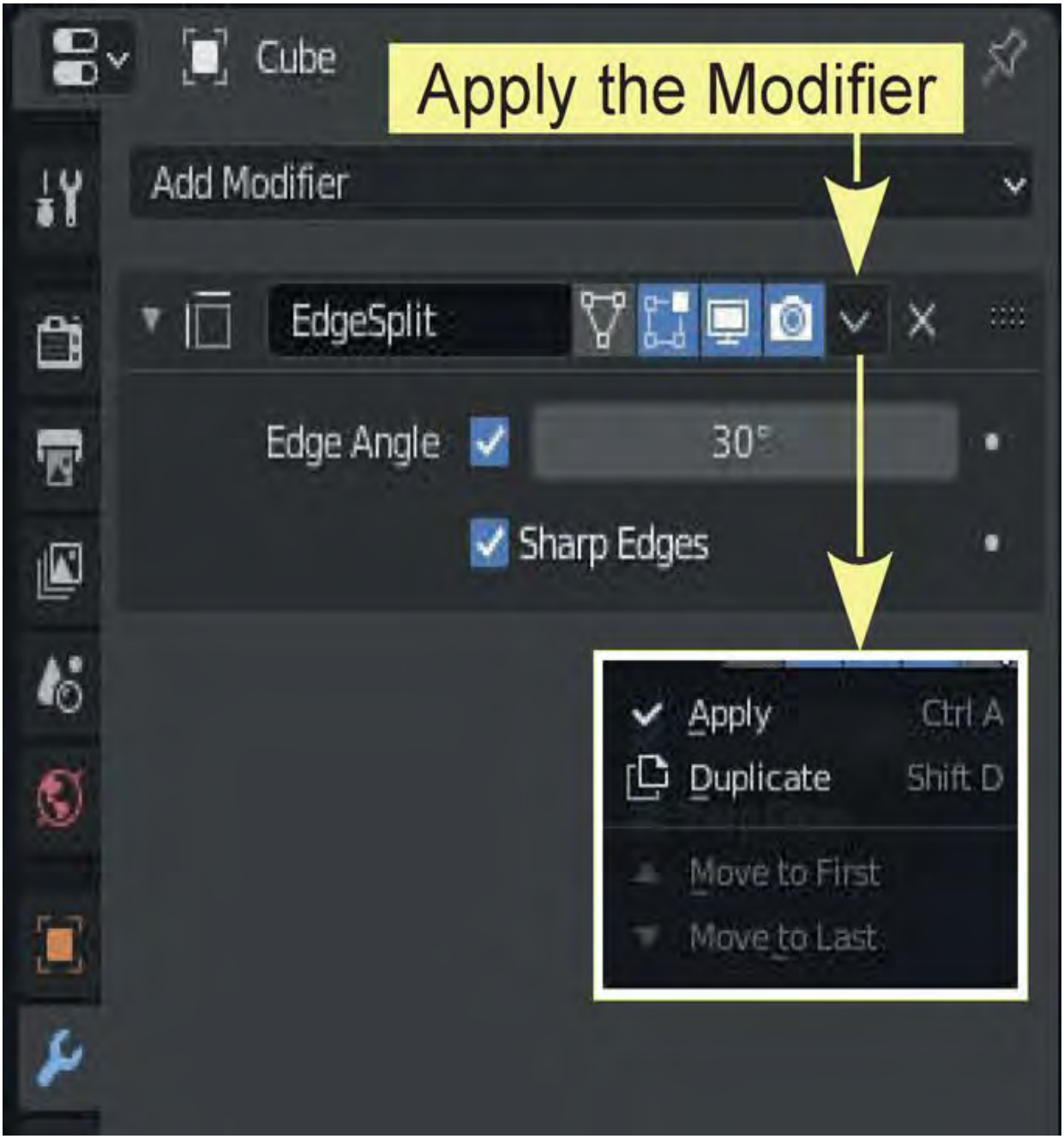


Figure 8.27

Remember the selection options in the Viewport Header (Chapter 5 – 5.2).

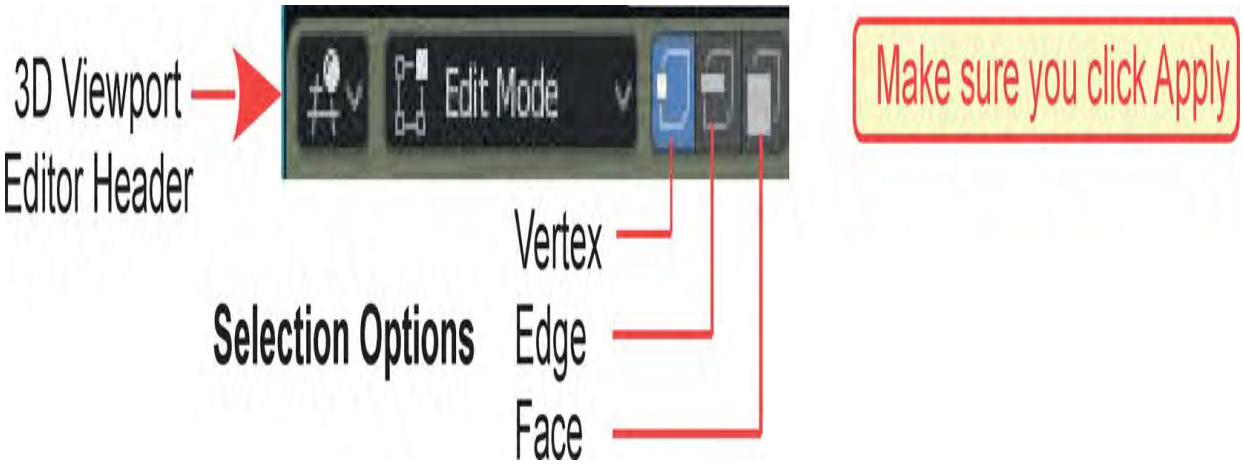


Figure 8.28

Make sure you click Apply

Choosing one of the Selection Options determines how the Edge Split Modifier operates.

After Applying the Modifier in Object Mode the Modifier is operable in Edit Mode.

Edge Split – Vertex

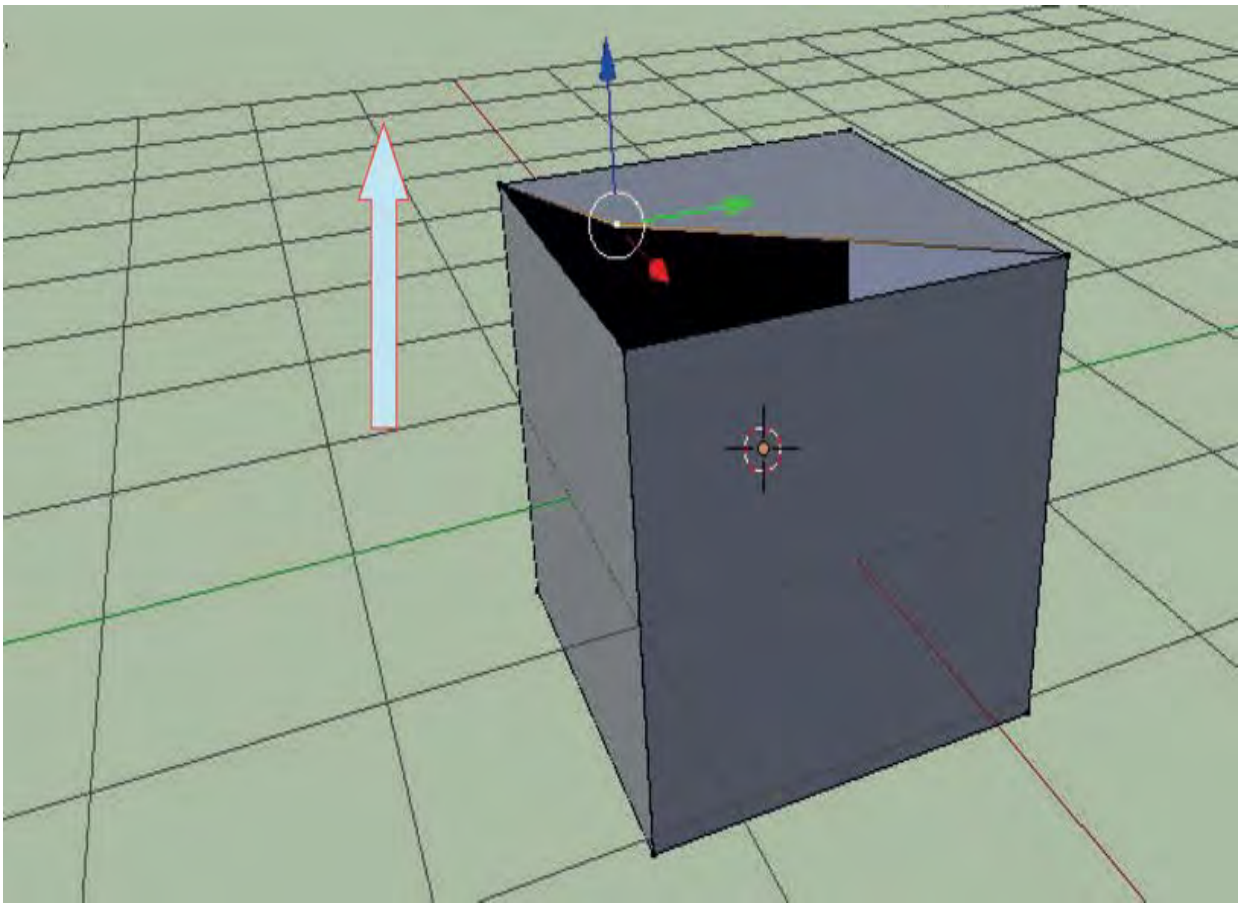


Figure 8.29

Selecting a **Vertex** will allow a corner to be lifted (moved.

Edge Split - Edge

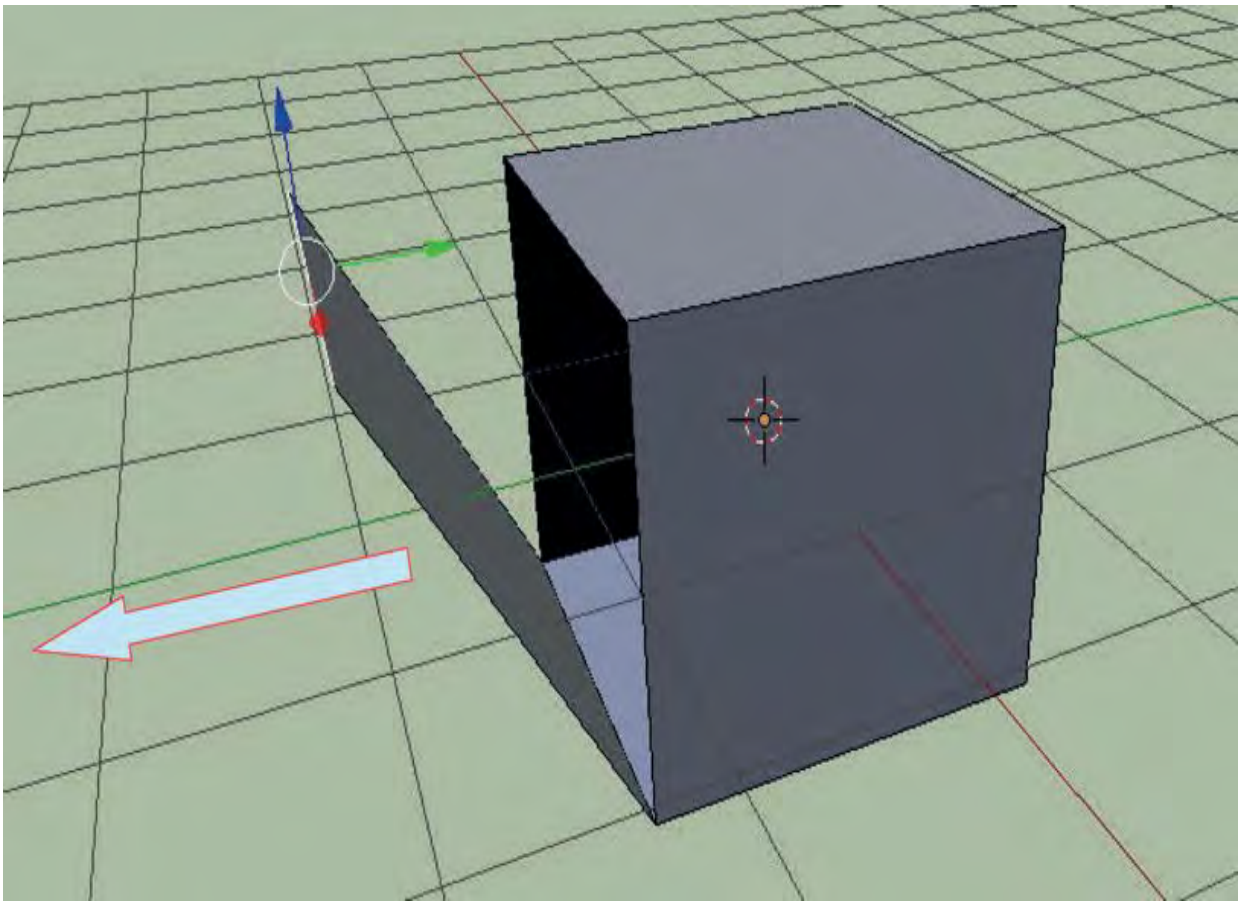


Figure 8.30

Selecting an Edge will open a face like the lid on a box (see also: **Rip Region Tool** Chapter 6 - 6.21)

Edge Split - Face

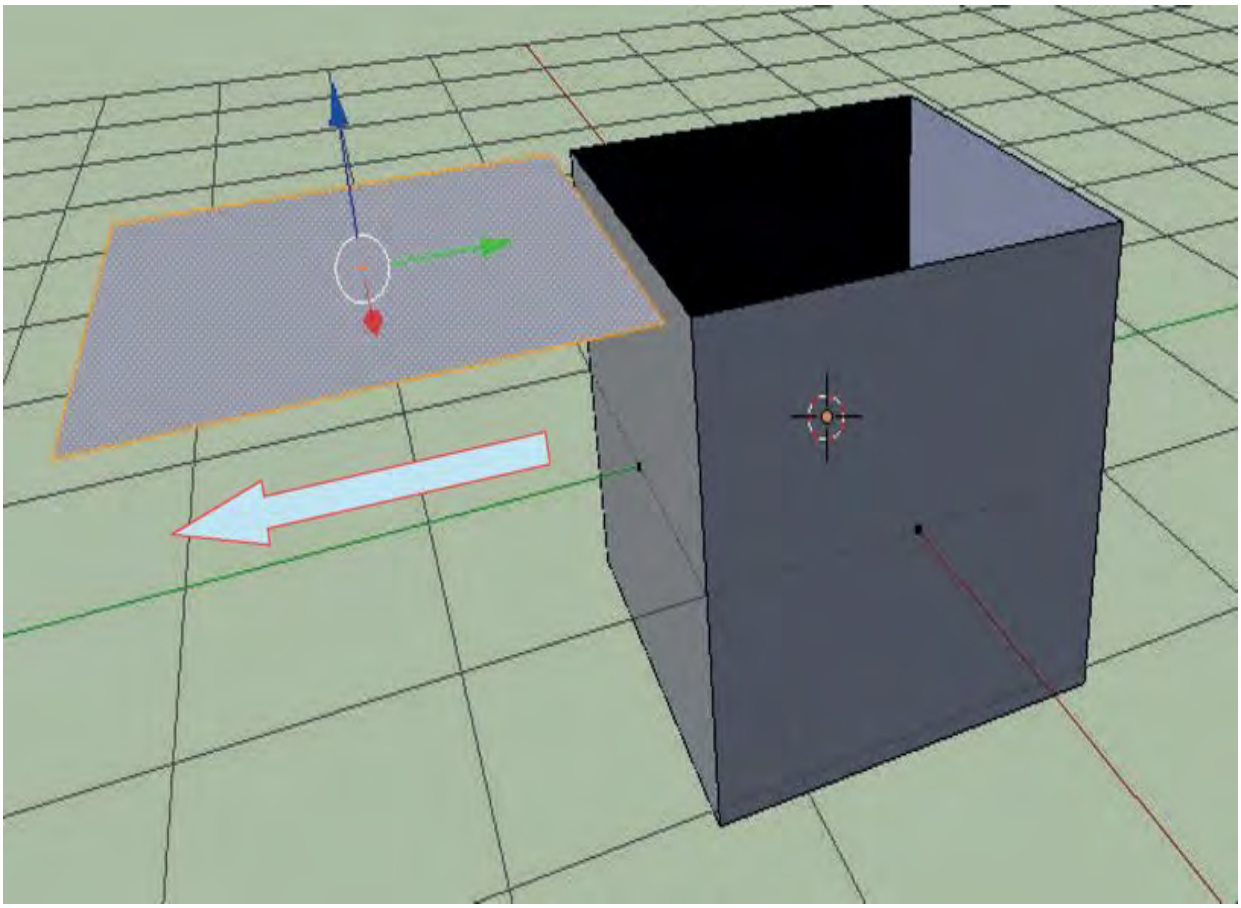
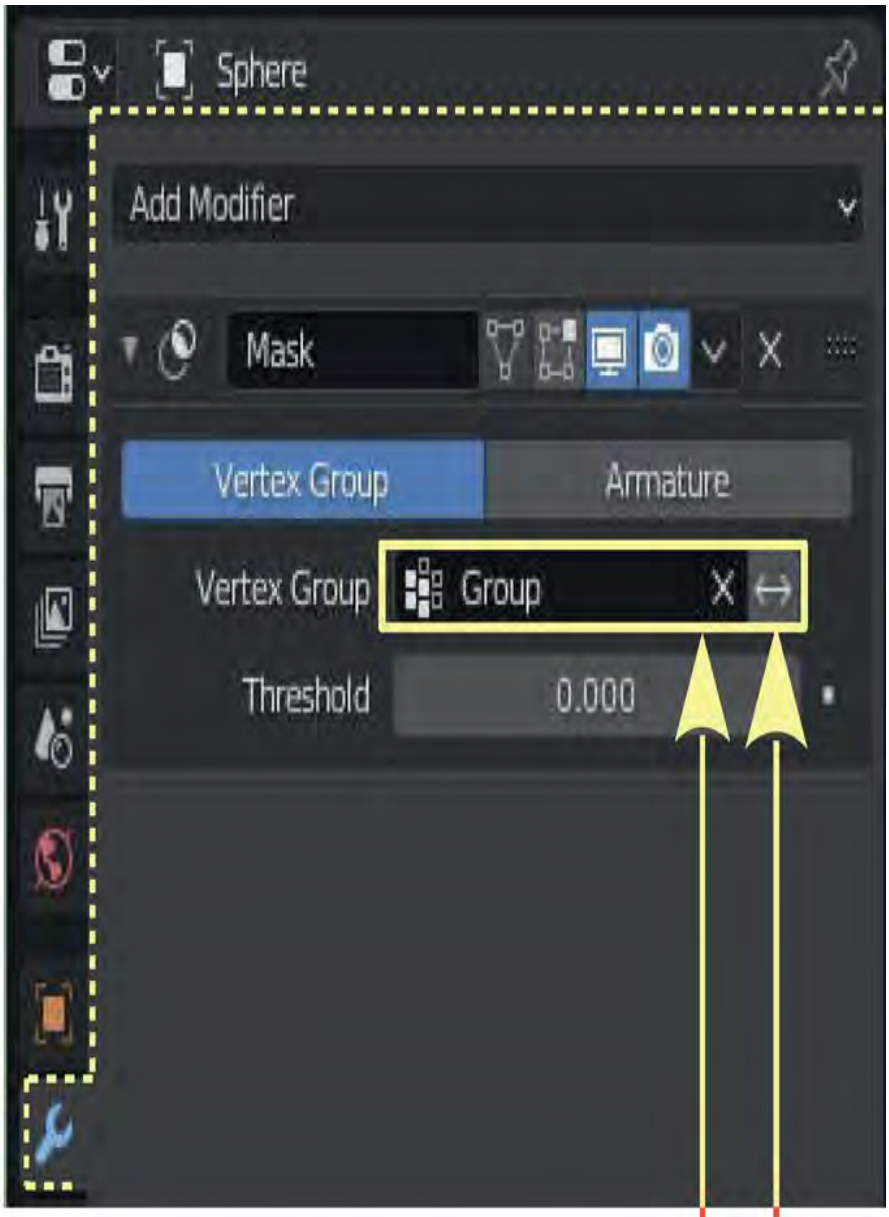


Figure 8.31

The Face remains part of the Object even though it is separated.

Select a **Face**. Use the **Move Tool Widget** to pull the Face away from the Cube.

8.8 Mask Modifier



While remaining in Edit Mode, Add a **Mask Modifier** to the Sphere and enter **Group**.

Click the double headed arrow to reverse the display (Figure 8.35)

Figure 8.32

The **Mask Modifier** allows you to limit what part of a mesh displays in the 3D View Editor or what part Renders. The part of the mesh is defined by a **Vertex Group** (see Chapter 5 – 5.9). Add a **UV Sphere** to the Scene. In **Edit Mode** select Vertices (Figure 8.33). Leave the Vertices selected and create a Vertex Group. By default the Vertex Group will be named **Group**.

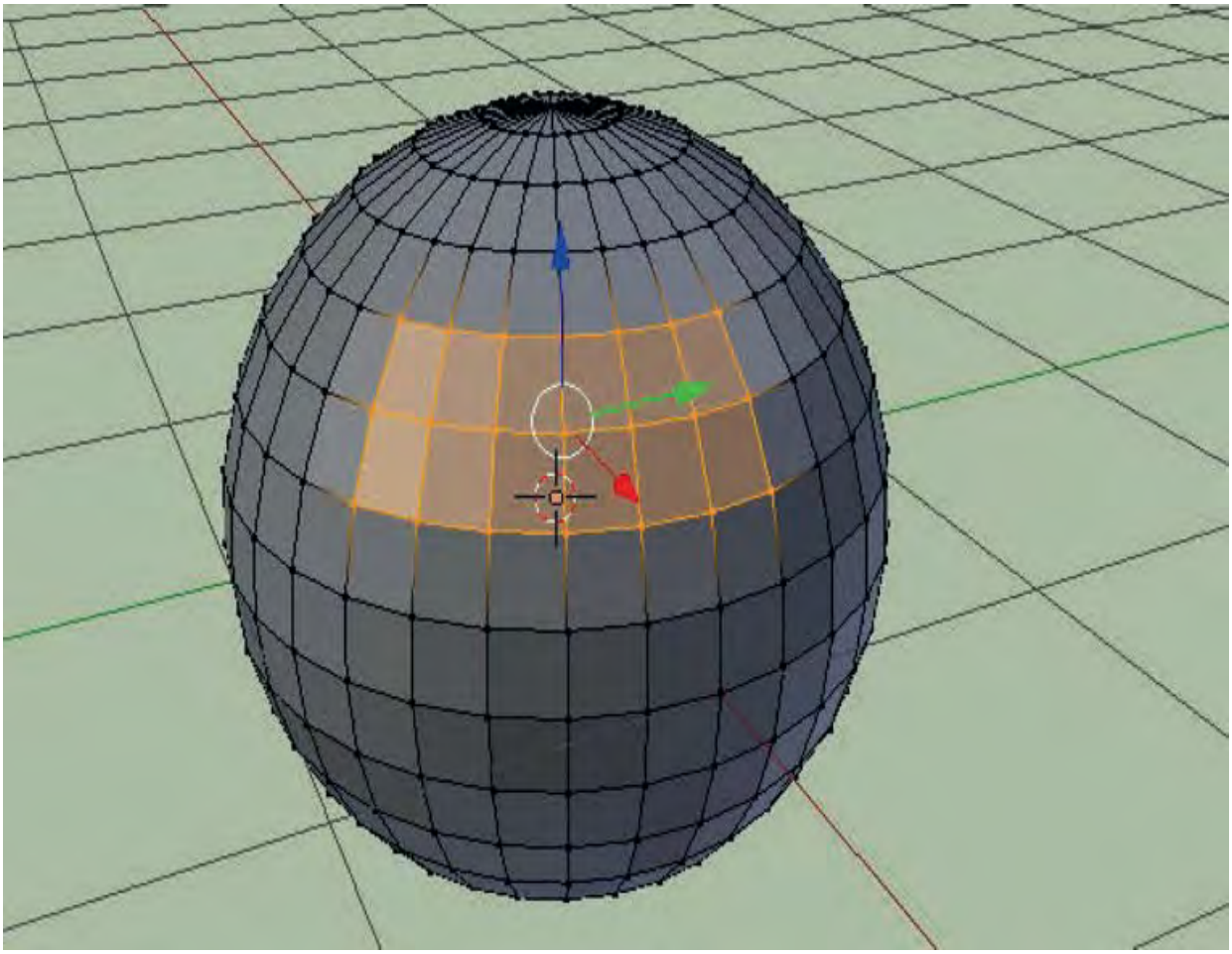


Figure 8.33

Tab into **Object Mode** to see only the part of the Sphere defined by the Vertex Group displayed (Figure 8.34). No need to Apply the modifier unless you want this to be permanent.

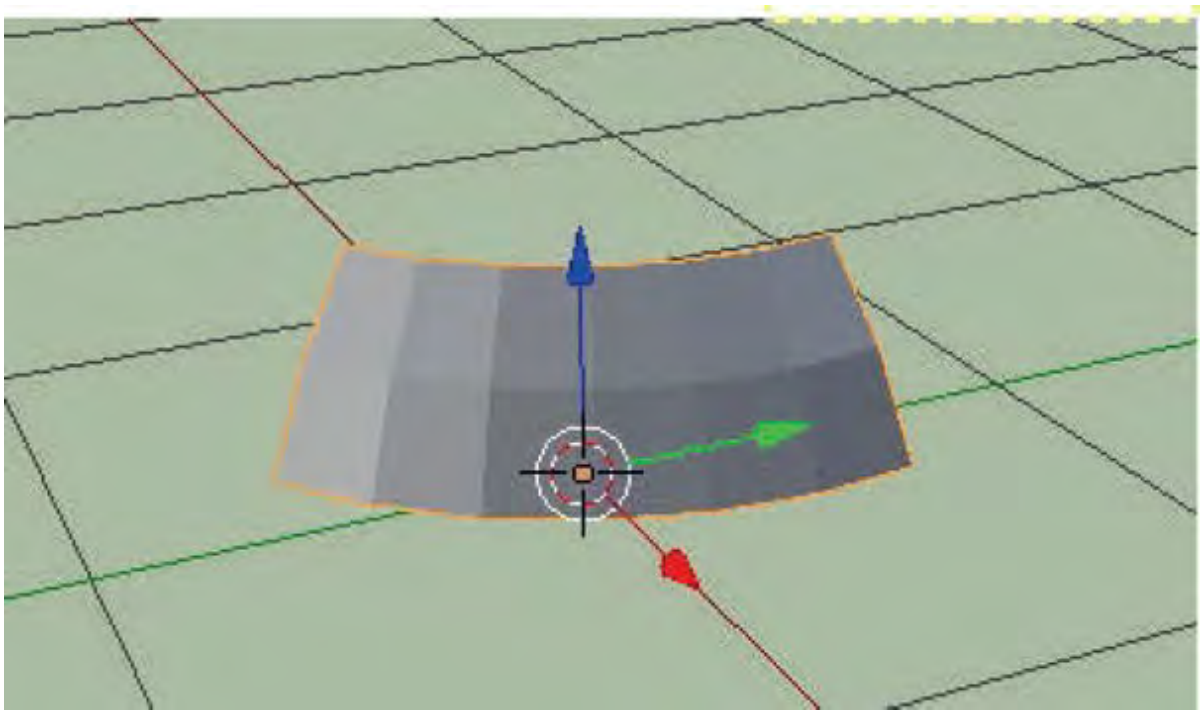


Figure 8.34

To reverse the display, that is, display the whole Sphere less the section defined by the Vertex Group, click the double headed arrow in the Modifier oabek (Figure 8.32, 8.35).

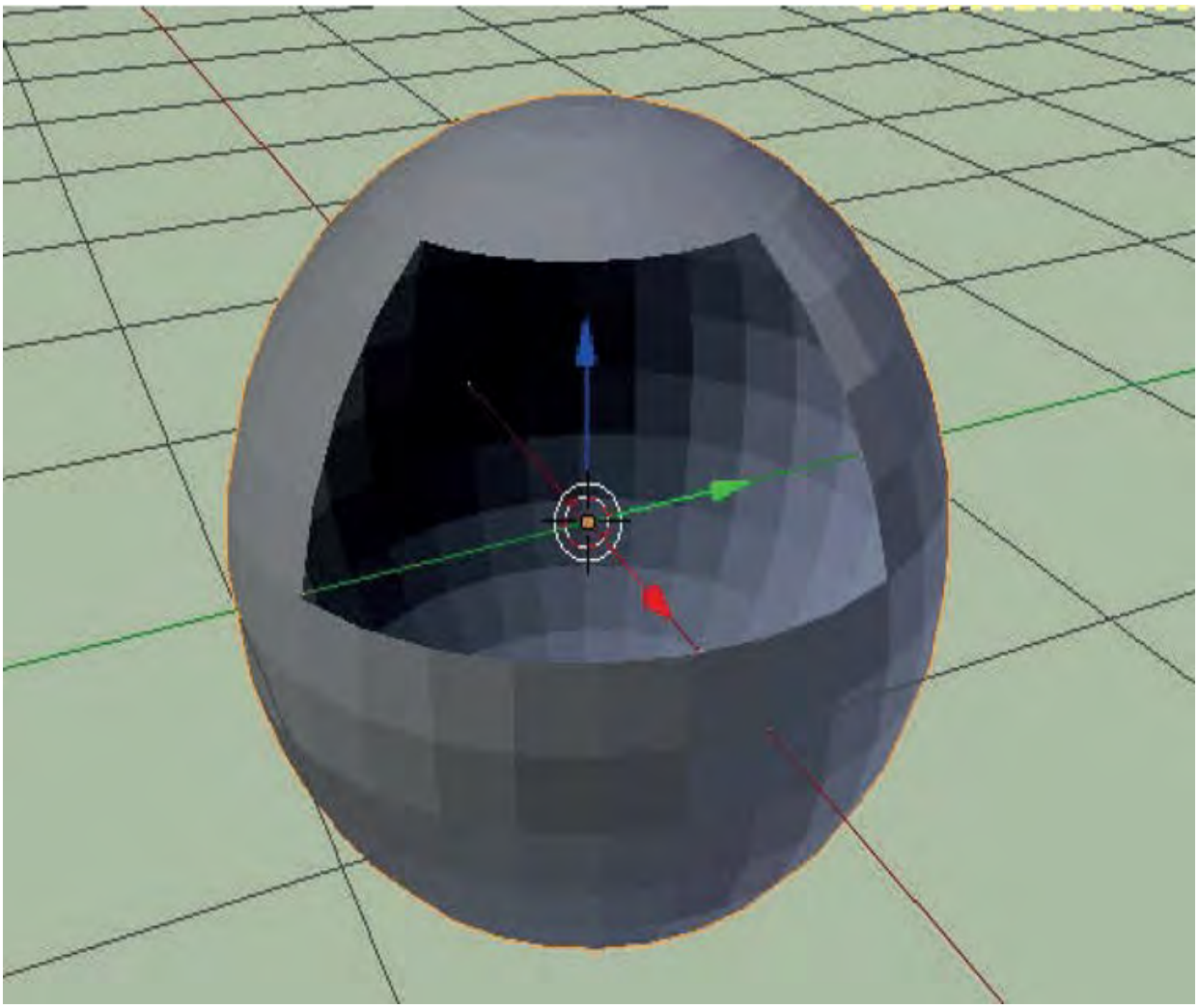


Figure 8.35

By using the modifier, visibility can be controlled without removing any Vertices from the UV Sphere.

8.9 Mirror Modifier

The **Mirror Modifier** allows the construction or deformation of a mesh on one side of a centre point to be duplicated (mirrored) on the opposite side.

Add a **UV Sphere** to the Scene in **Top Orthographic View**. Tab to **Edit Mode**, deselect the Vertices. **Have Toggle X Ray on,** then B key (Box Select) and drag a rectangle to select one half of the Sphere's Vertices (Figure 8.36, 8.37). Press **X Key** to delete the selected Vertices (Figure 8.38).

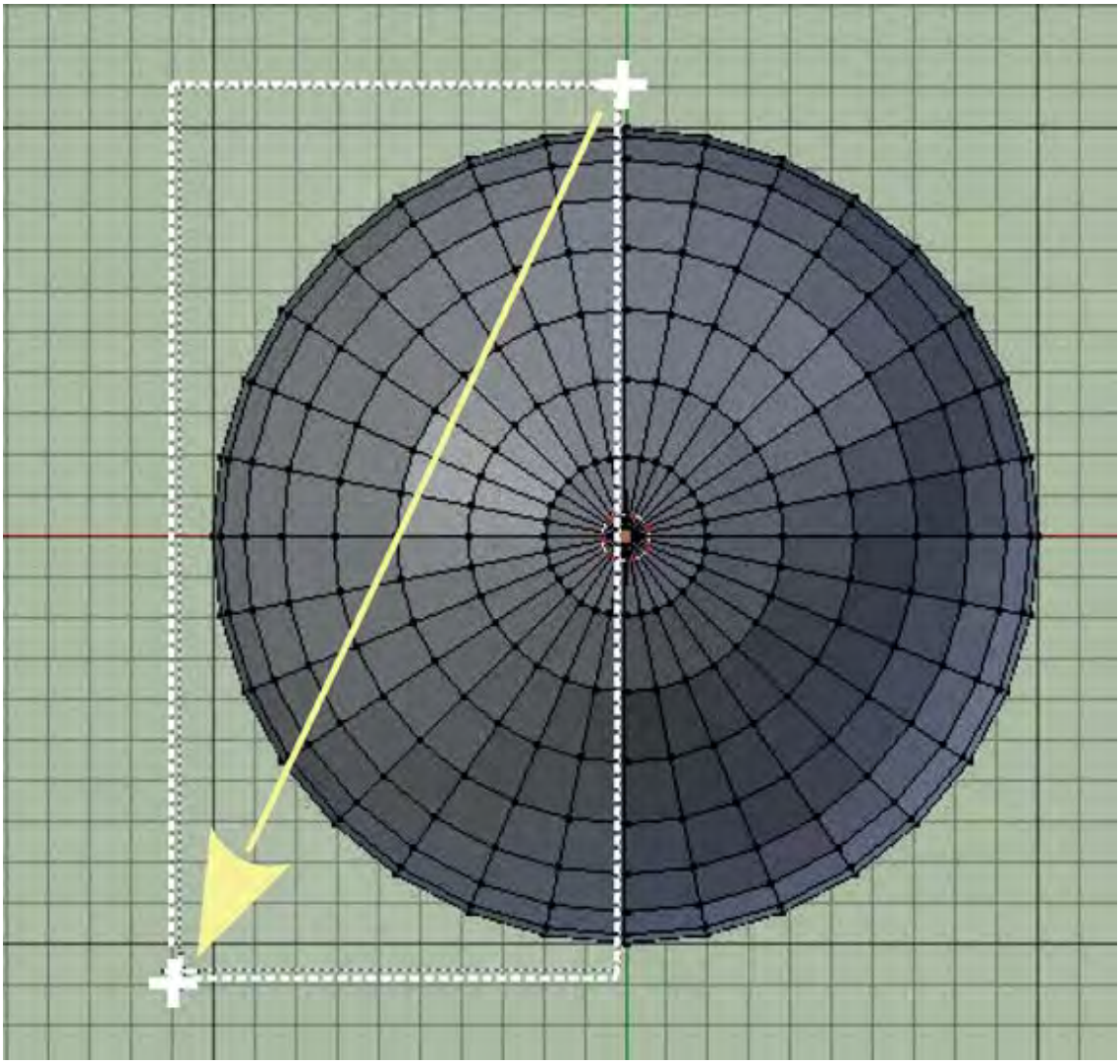


Figure 8.36

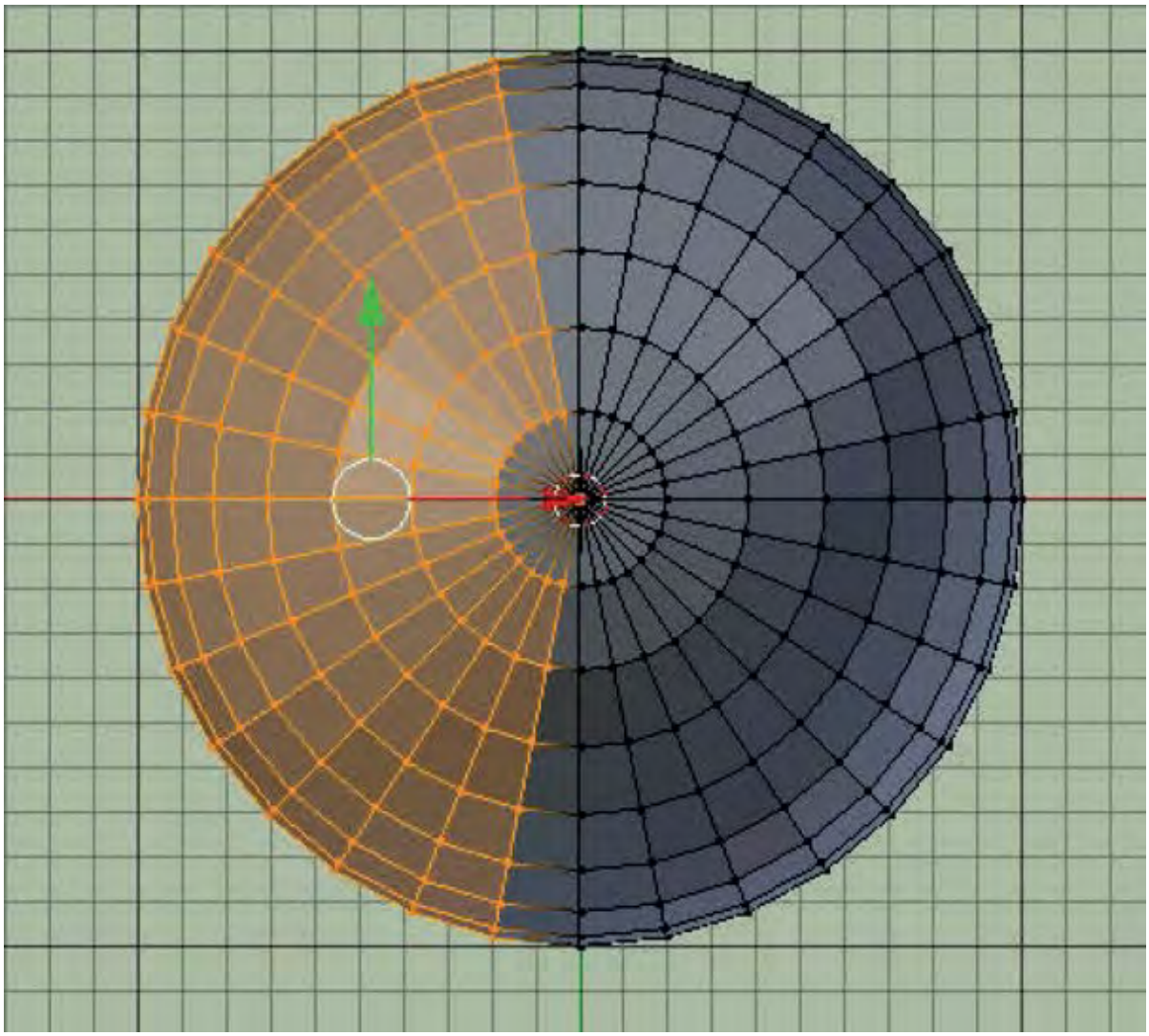


Figure 8.37

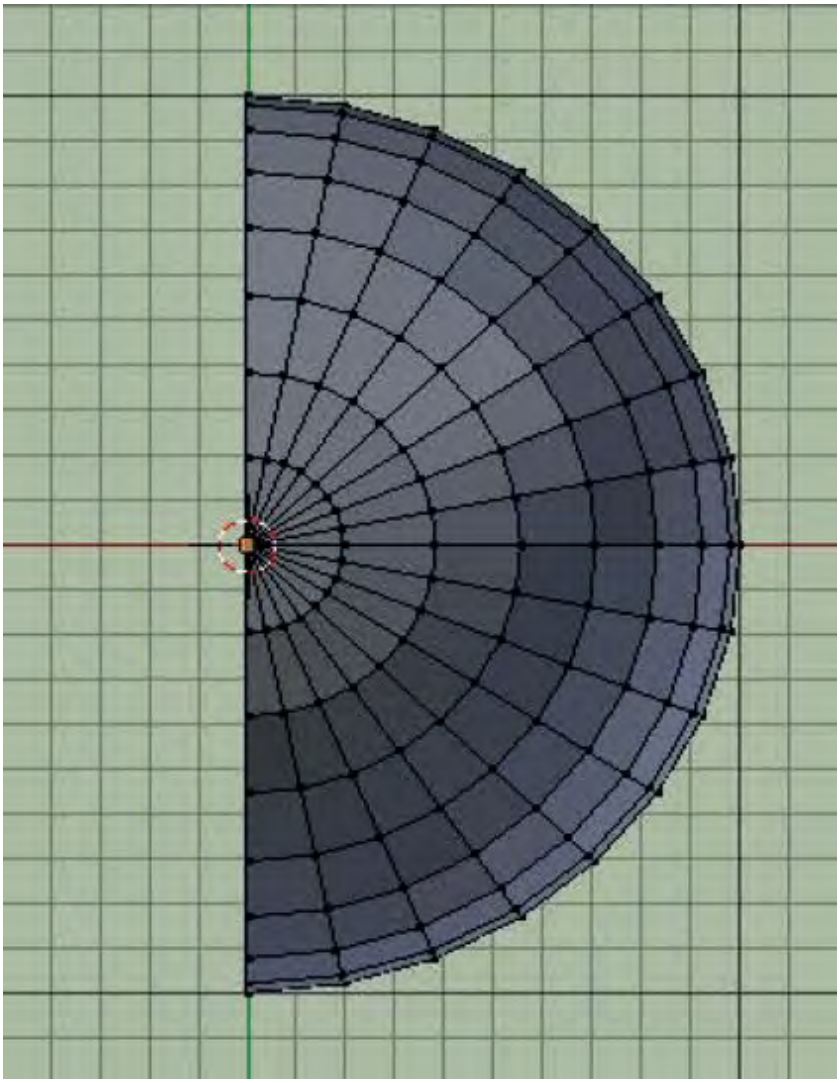


Figure 8.38

Tip: Don't forget to turn **Toggle X-Ray** on in the 3D Viewport Editor Header before dragging the rectangle.

In **Object Mode** (with the half sphere selected), in the **Properties Editor, Modifier buttons** add a **Mirror Modifier** (Figure 8.39 over).

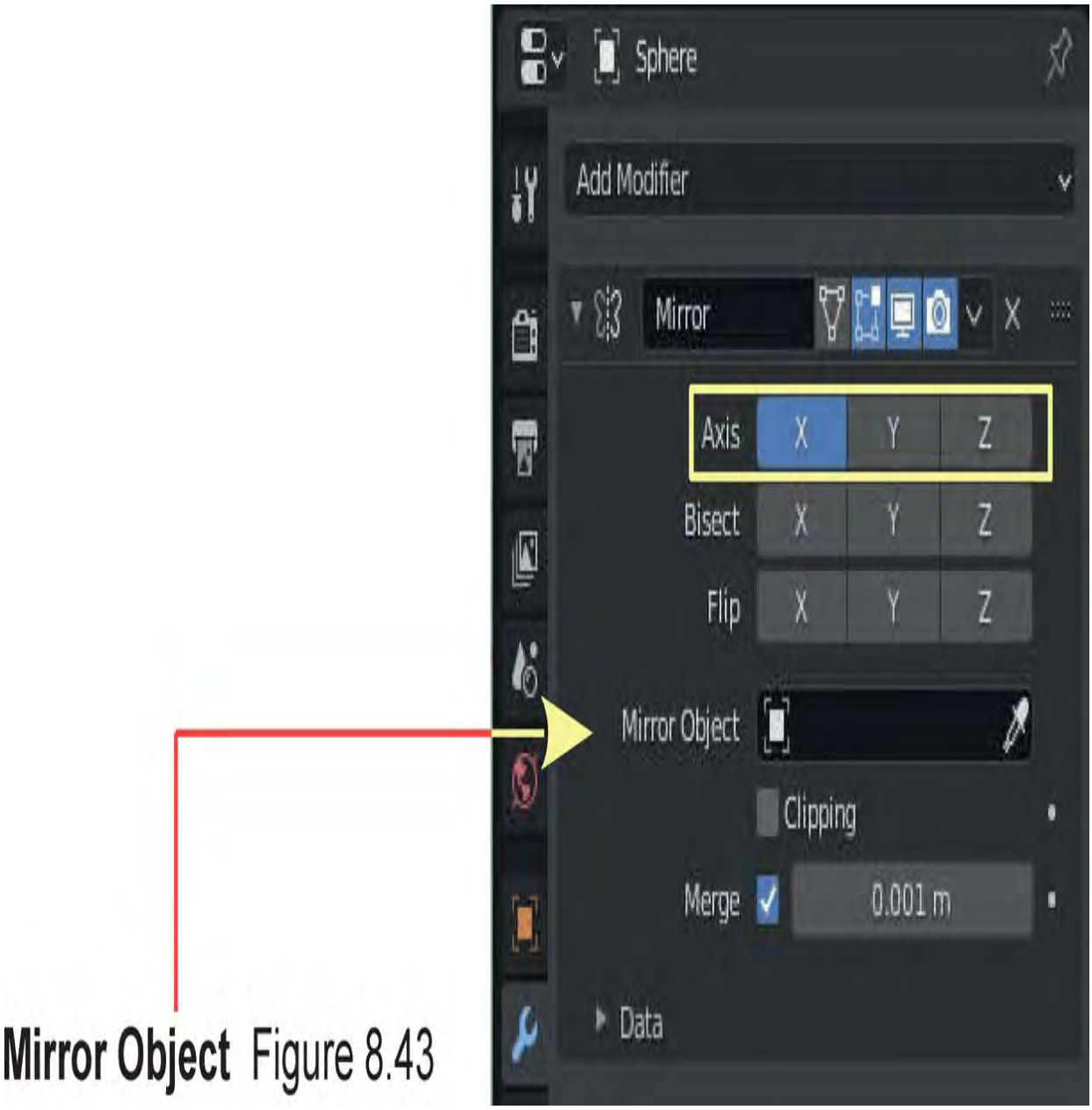


Figure 8.39

The deleted half of the UV Sphere will be reinstated in the 3D View Editor (Figure 8.40).

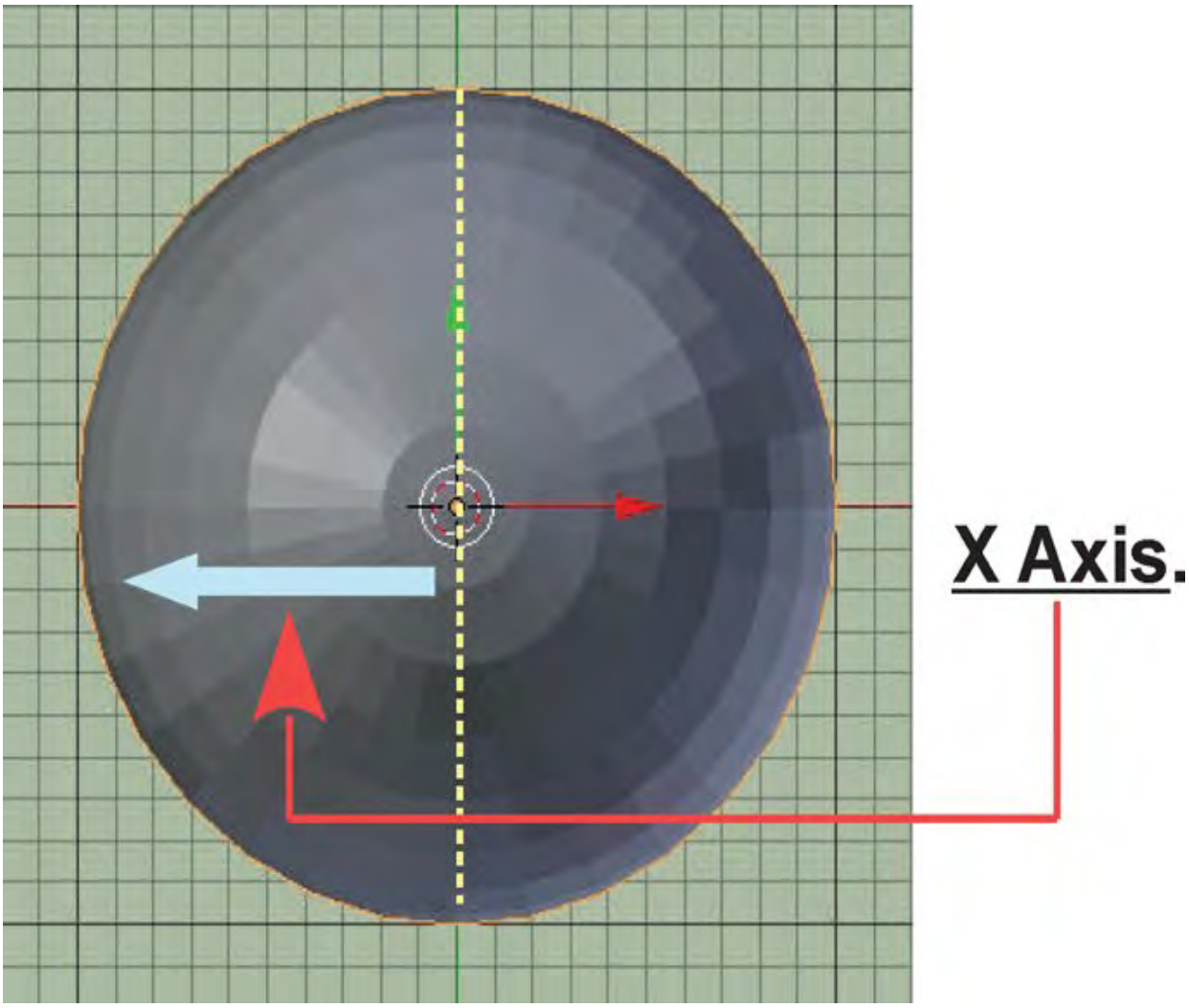


Figure 8.40

In **Edit Mode** you will see that vertices exist only on one side of the sphere (Figure 8.41).

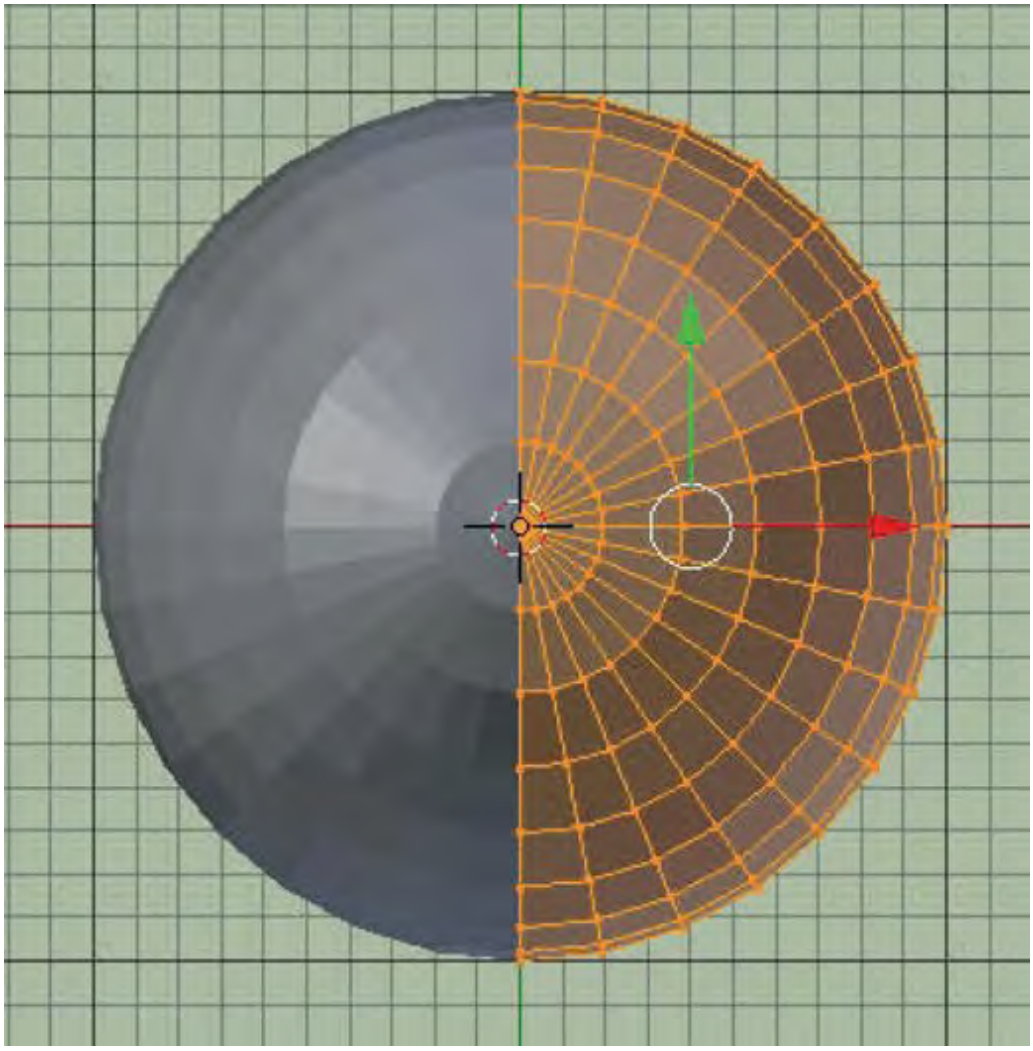


Figure 8.41

Select and Translate a single Vertex and you see that the mesh on the opposite side is duplicated (mirrored) (Figure 8.42).

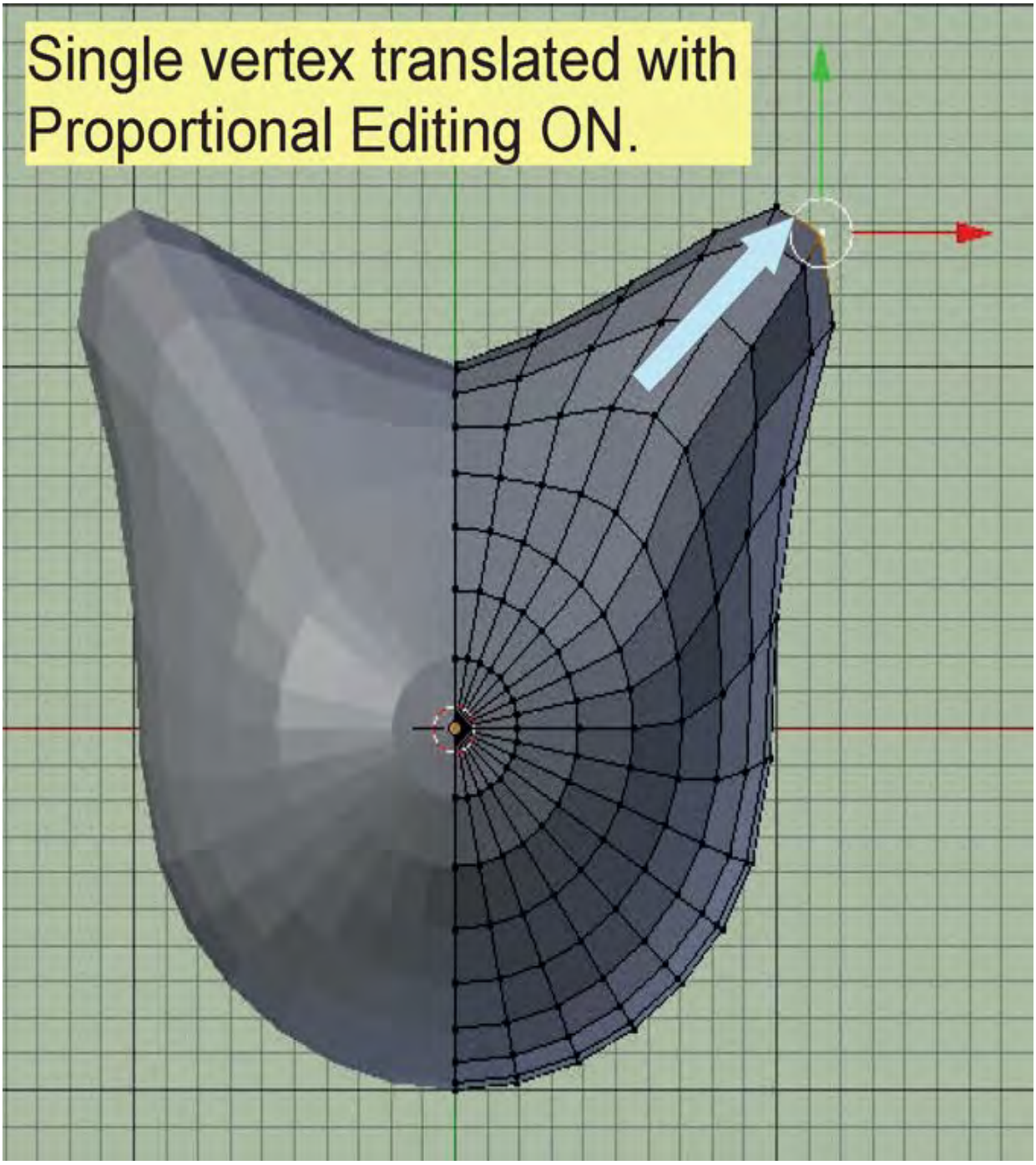


Figure 8.42

When the Modifier is applied (click **Apply** in the Modifier Panel) Vertices are created on the mirrored side.

In the center of the Mirror Modifier panel you will see **Mirror Object**. Placing an Object in the Scene and entering the name of this Object in the Mirror Object Bar causes the mirror to be about the centerline of the new Object instead of the centerline of the original Object (Figure 8.43).

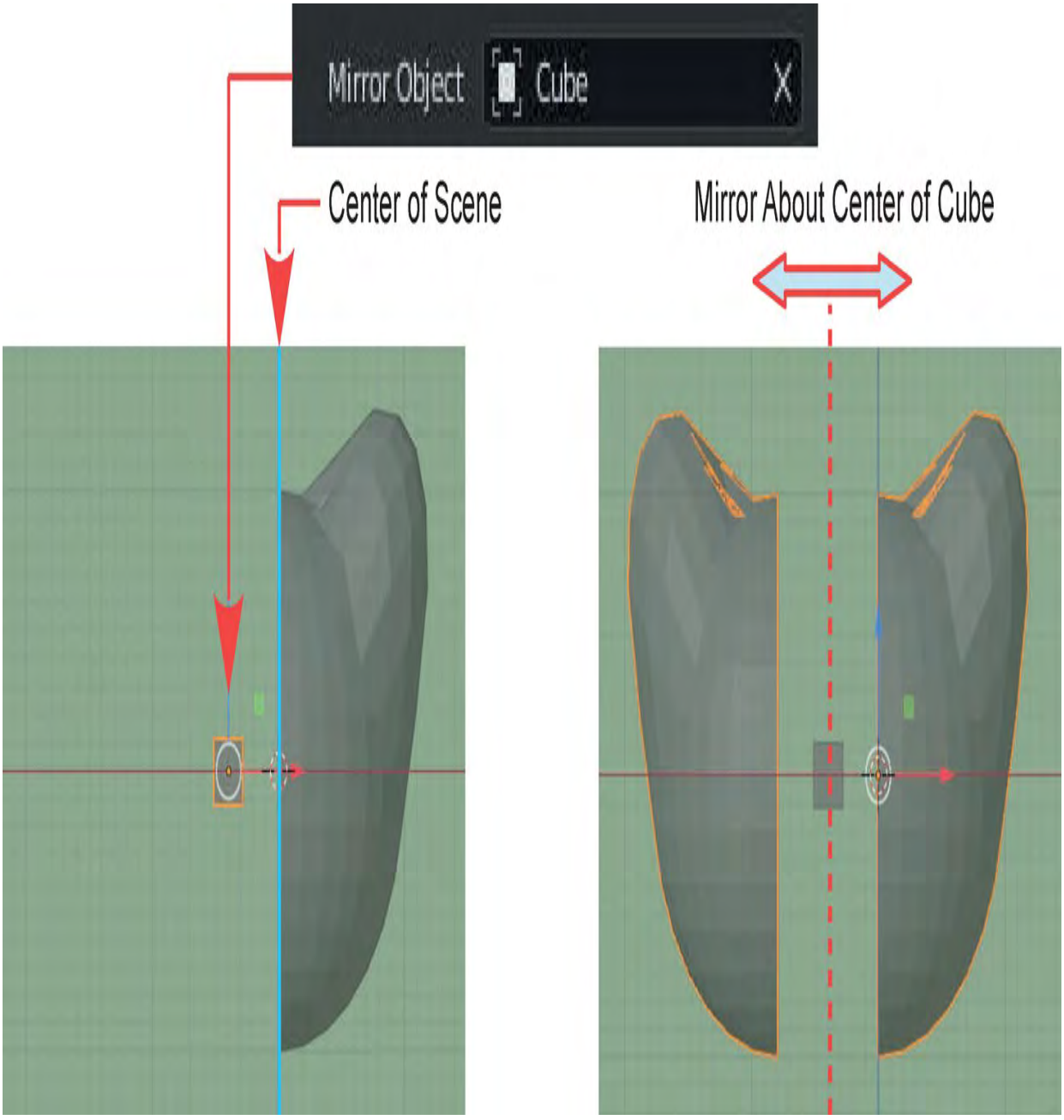


Figure 8.43

8.10 Multiresolution Modifier

The **Multiresolution Modifier** is designed to be used with the **Sculpt Tool** (Chapter 11 –11.4, 11.5).

8.11 Remesh Modifier

The **Remesh Modifier** allows you to recalculate how a Mesh Surface is constructed. Some basic mesh shapes do not provide sufficient Vertices, Edges and Faces to allow detailed modeling.

To demonstrate, use a Cylinder Object to cut a hole through the default Cube Object by applying a **Boolean Difference Modifier (to the Cube Object)**.

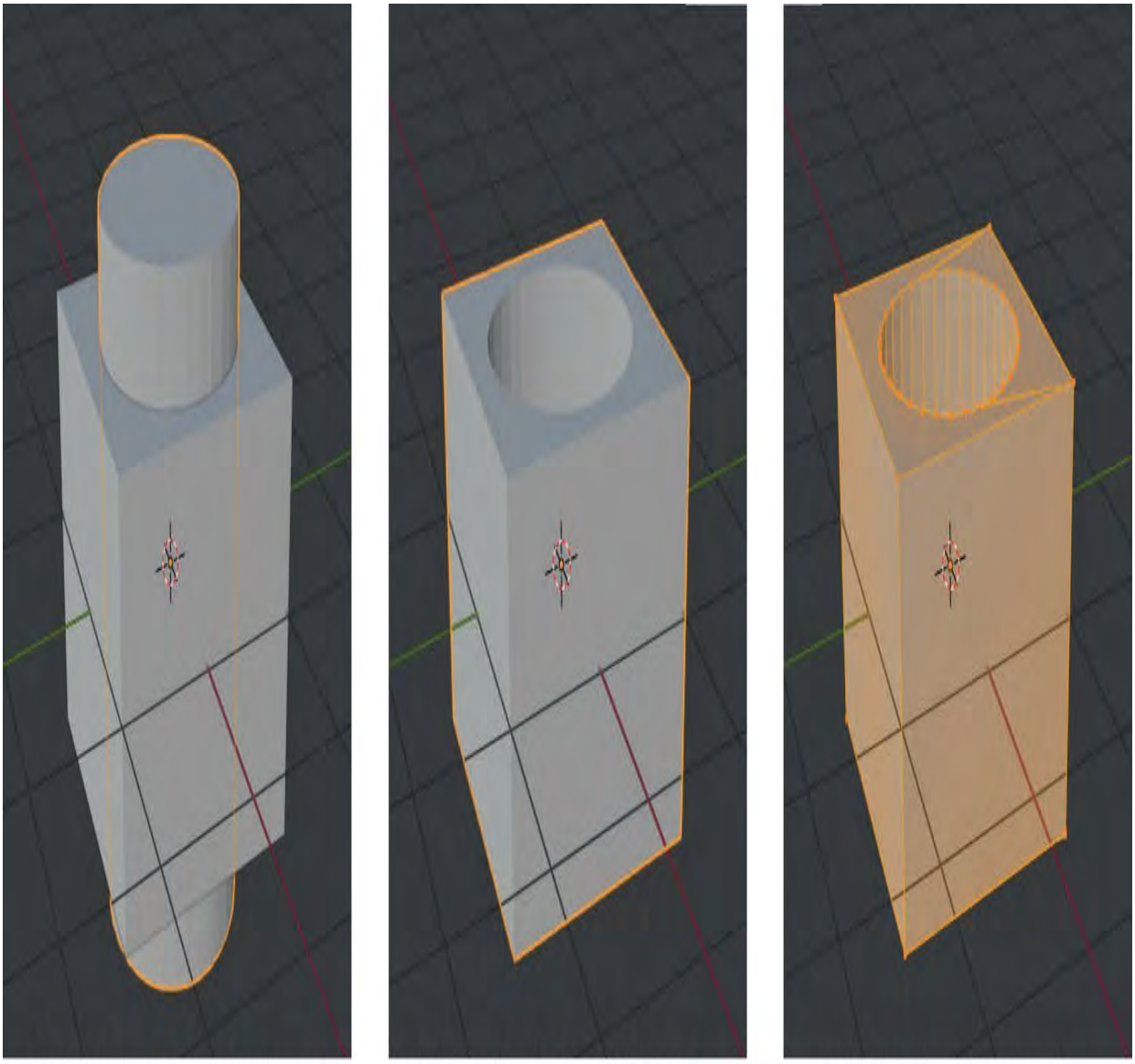

Cylinder scale down and extended on the Z Axis.
The Boolean Difference Modifier Applied.
Edit Mode showing the Mesh construction

Figure 8.44

When the Boolean Modifier is applied you see a minimal mesh construction which limits any detailed modeling. To increase the Vertex count apply a **Remesh Modifier.**

Remember: To apply a Modifier to an Object it must be in Object Mode.

With the Cube selected in Object Mode add a **Remesh Modifier.** In **Object Mode** you see a deformation at the rim of the hole. **Note: The default Remesh Mode is Voxel. Change the Mode to Sharp.**

Figure 8.45

With the default values, when the Modifier has been Applied (click the Apply button) the Mesh surface of the Cube in Edit Mode shows a significant increase in the number of Vertices, Edges and Faces (Figure 8.46).

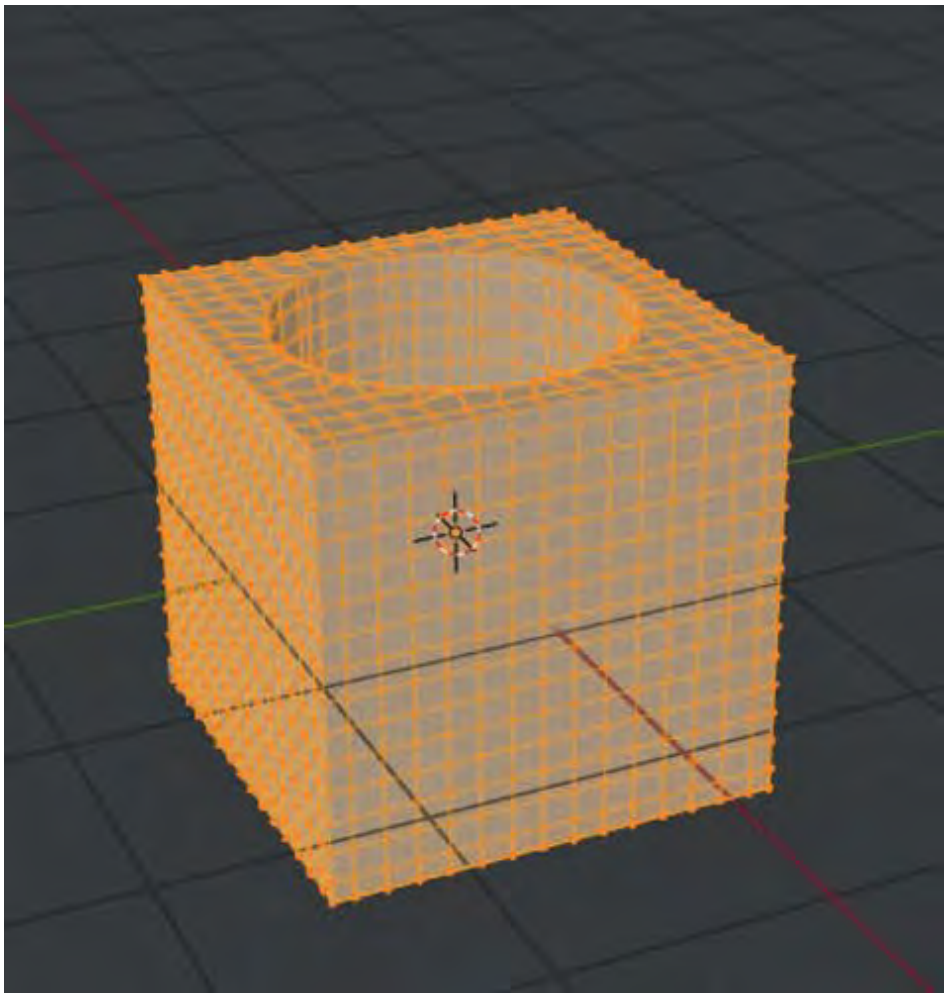


Figure 8.46

When the **Remesh Modifier** is Applied the Modifier Panel is cancelled. To adjust the Octree and Scale values press Ctrl + Z Key to undo and step back through the operations until the panel is reinstated. Increasing the Octree

Depth significantly increases the number of Vertices on the Mesh. Even increasing from 4 to 5 has a dramatic effect.

Be Warned: Increasing to 6 - 7 - 8 will exponentially increase the Vertex Count and seriously affect computer speed.

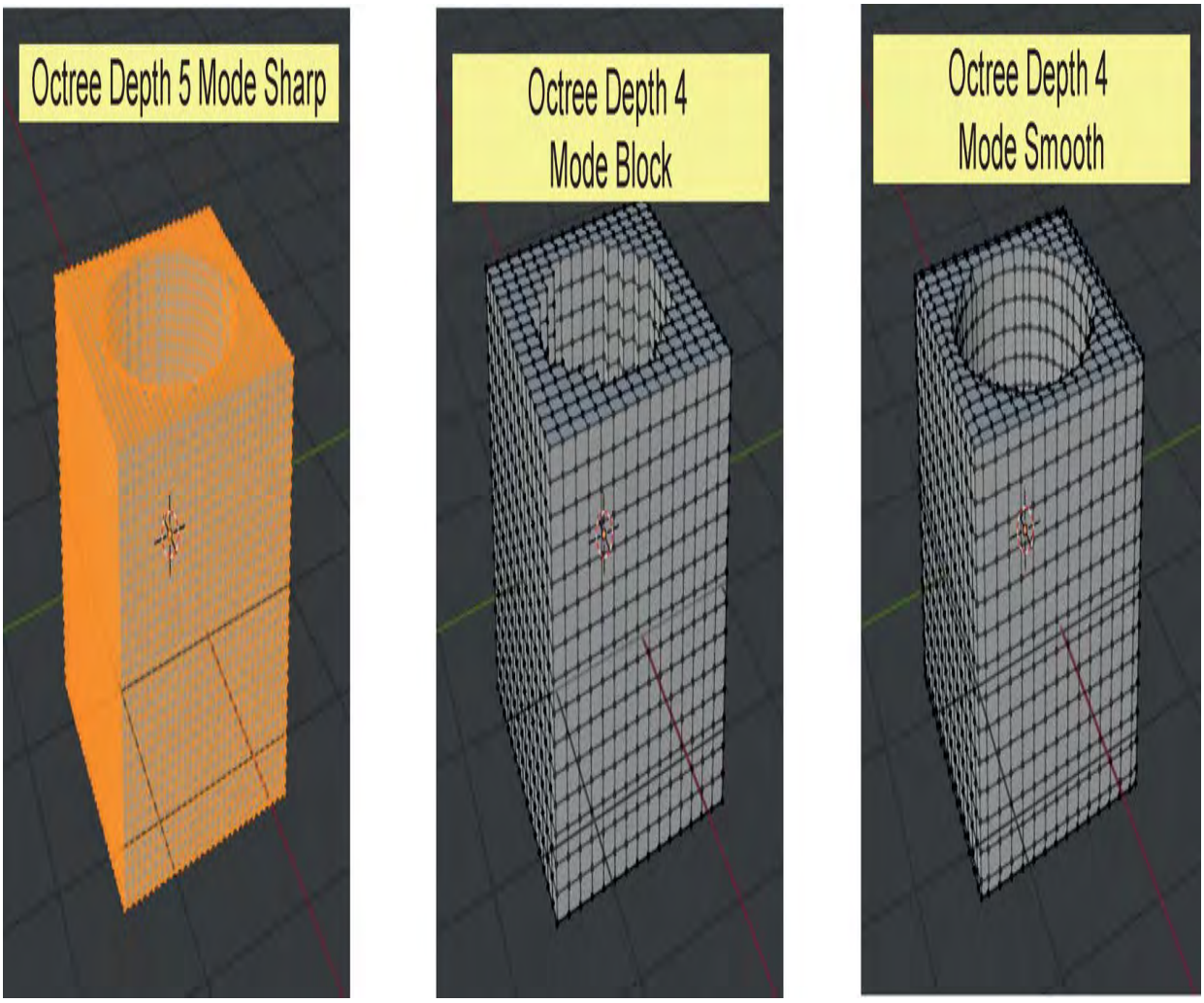


Figure 8.47

8.12 Screw Modifier

The **Screw Modifier** generates a spiral shape by revolving a profile or an Object around an Axis defined by an Axis Object. To demonstrate, construct a spiral from the a Plane Object revolved around an Axis defined by a Cylinder.

The Plane, by default is entered into the Scene in the horizontal plane. Rotate (R Key + X Key + 90) to stand the Plane on edge then move it back along the X Axis of the Scene. Enter a Cylinder Object, scaled down and scaled up on the vertical Z Axis (Figure 8.49).

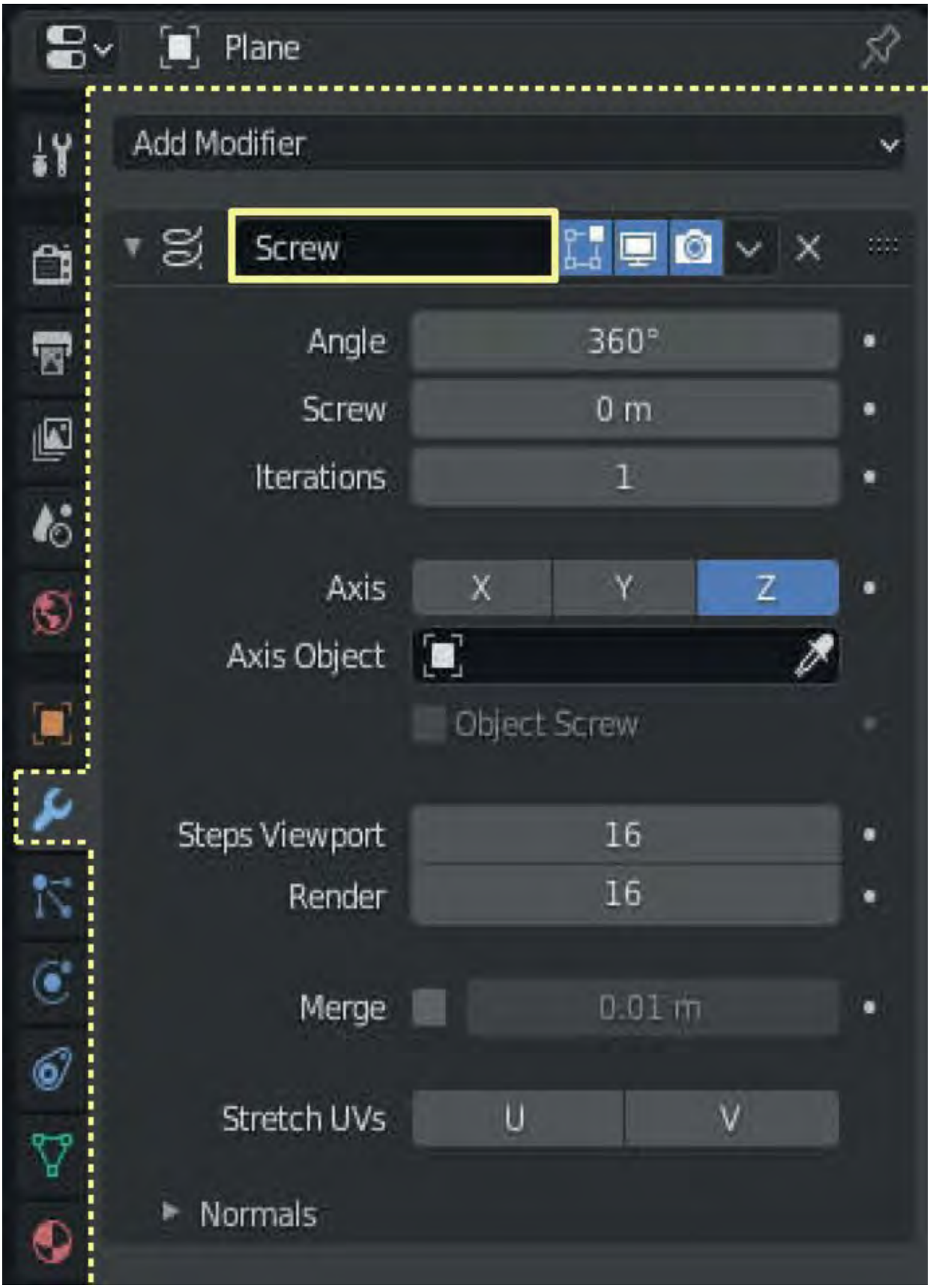

Plane
Add Modifier
Screw
Angle
360°
Screw
0 m
Iterations
1
Axis
X
Y
Z
Axis Object
Object Screw
Steps Viewport
16
Render
16
Merge
0.01 m
Stretch UVs
U
V
Normals

Figure 8.48

With the Plane selected Add a Screw Modifier in the Properties Editor, Modifier buttons (Figure 8.48).

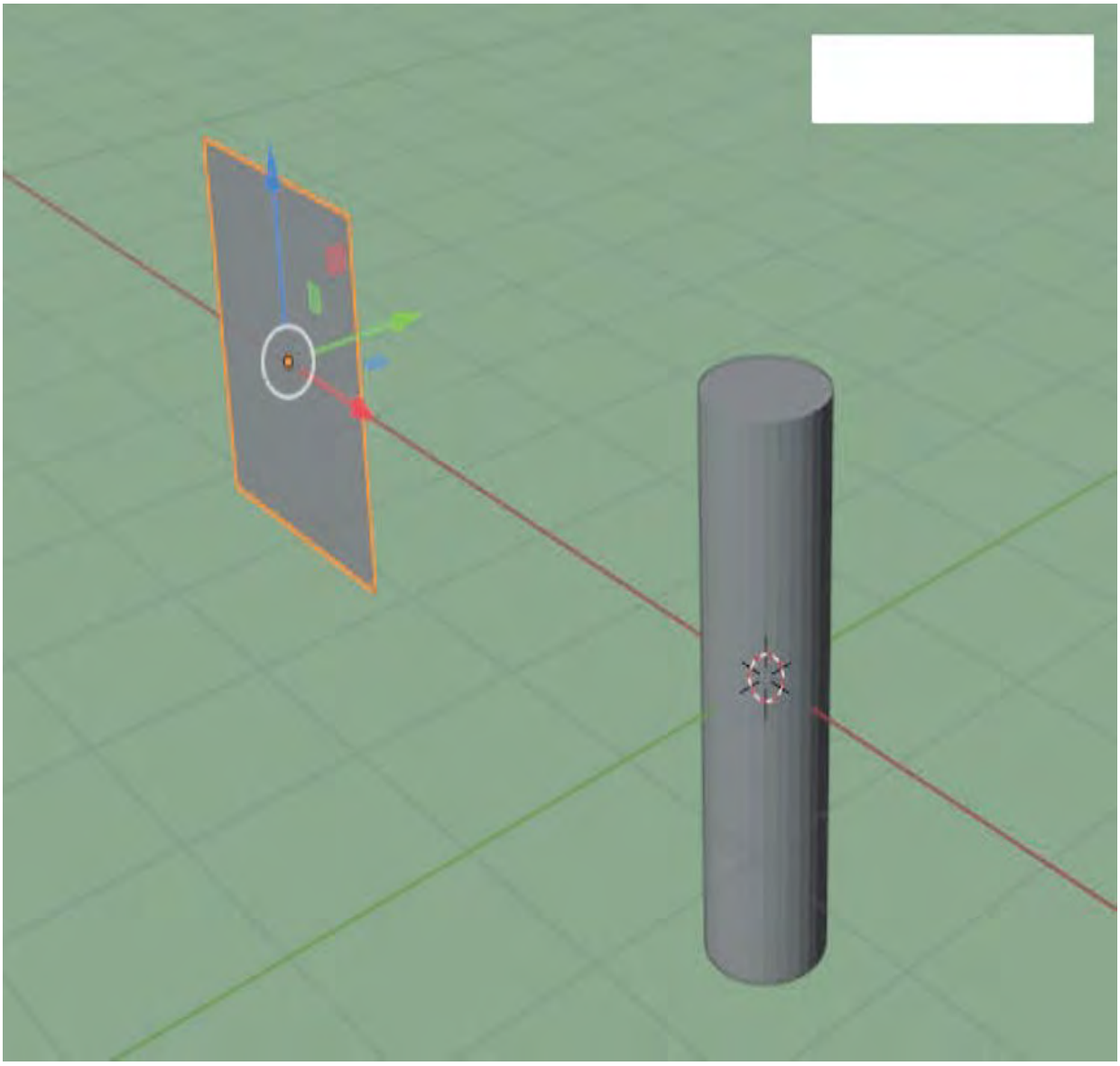


Figure 8.49

When the Screw Modifier is Added the Plane displays as a flat circular shape in the XZ Plane (Figure 8.49).

Without an Axis Object being entered in the Modifier panel the Plane is duplicated and revolved 360° about its own Y Axis (Figure 8.50). The degrees of revolution are defined by the Angle value in the Modifier panel. The number of duplications are defined Steps value in the Modifier panel (Note: there are different Steps values for the Viewport and Render. The default values are 16).

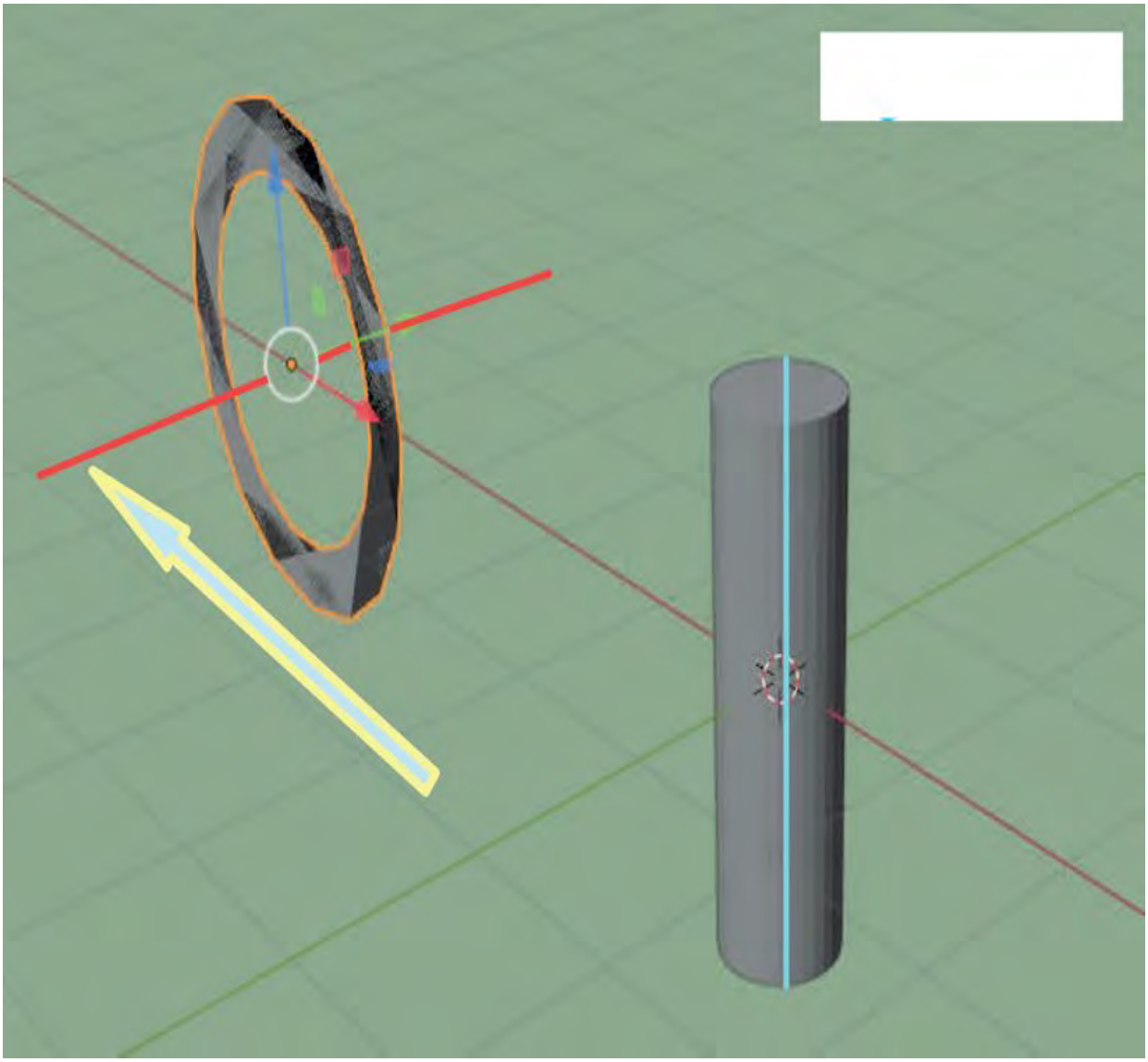


Figure 8.50

Figure 8.51 depicts the revolution of the Plane duplications.
This effect may be useful but it is not the Screw that is intended.

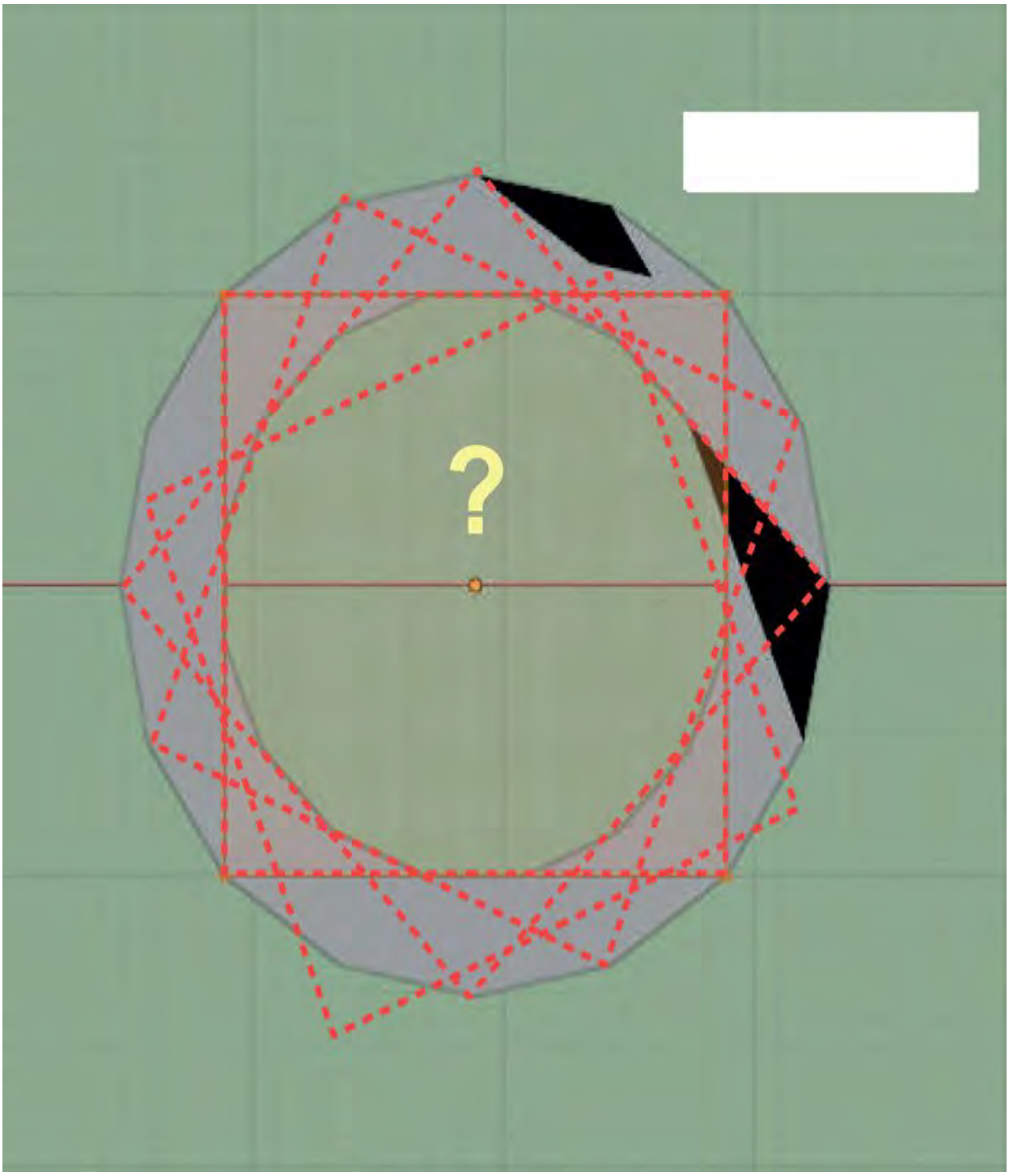


Figure 8.51

Enter **Axis Object: Cylinder** in the Modifier panel.

Change the 3D Viewport Editor to **Edit Mode**.

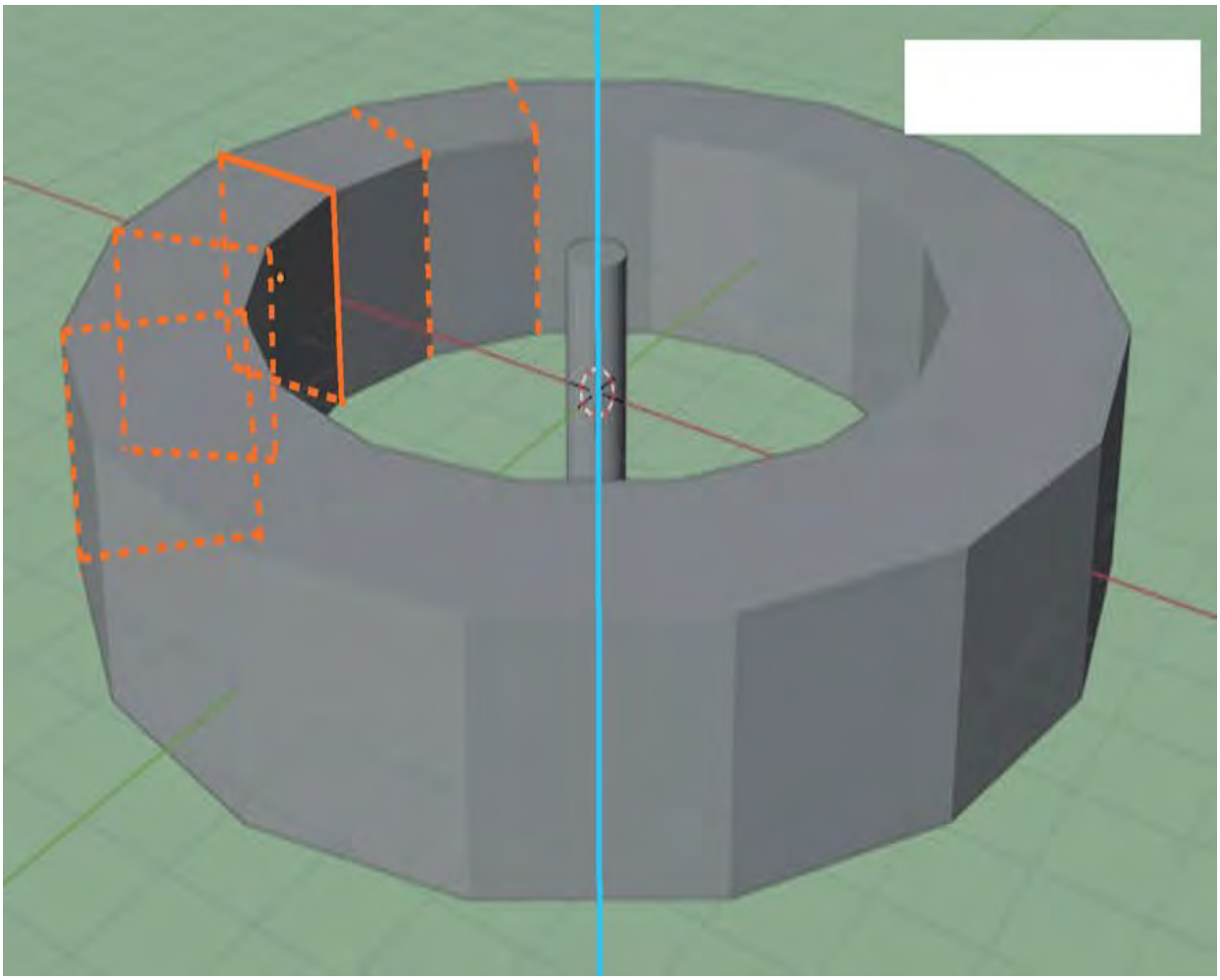


Figure 8.52

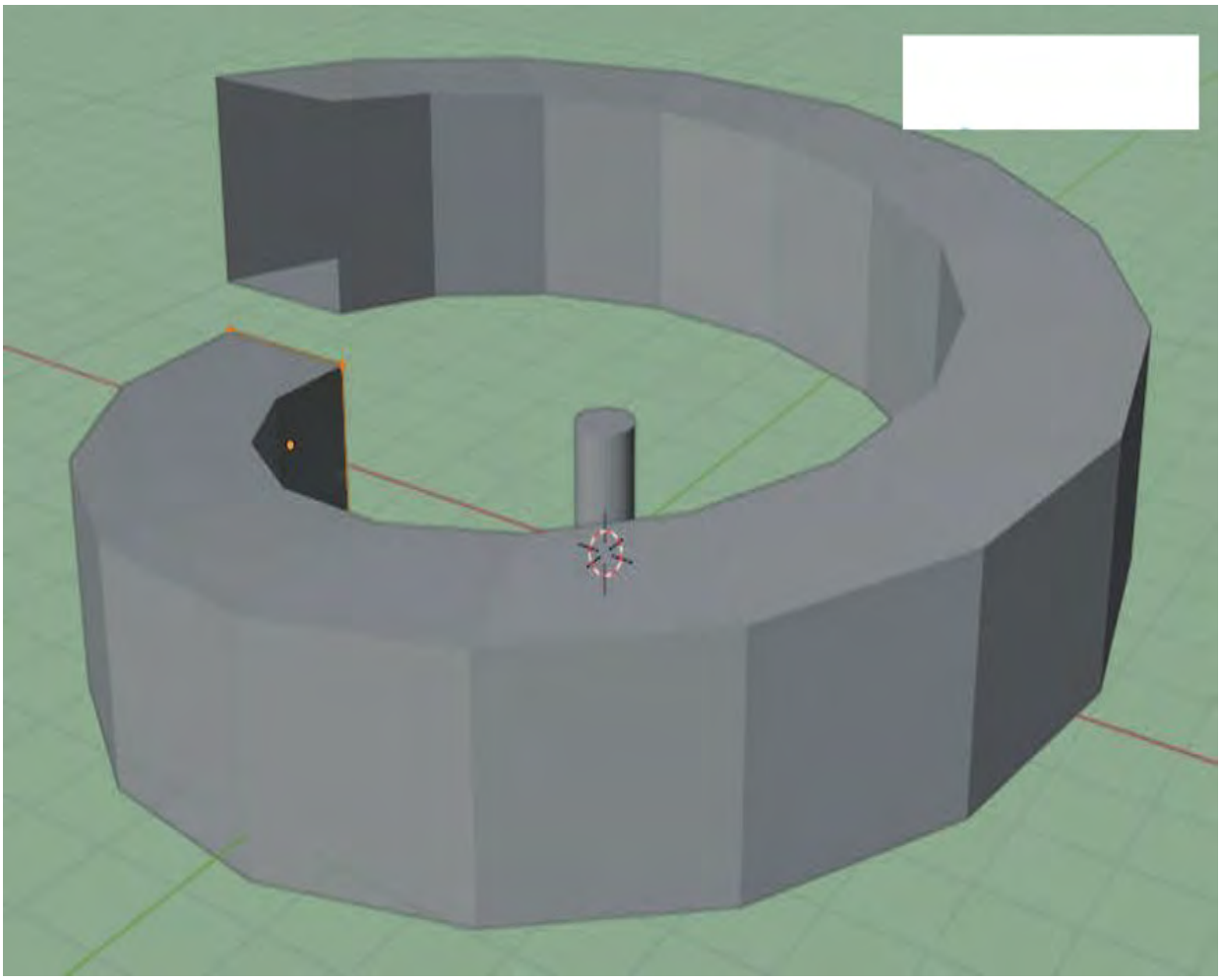


Figure 8.53

Increase the Iterations value in the Modifier panel to increase the number of coils. (Figure 8.54)

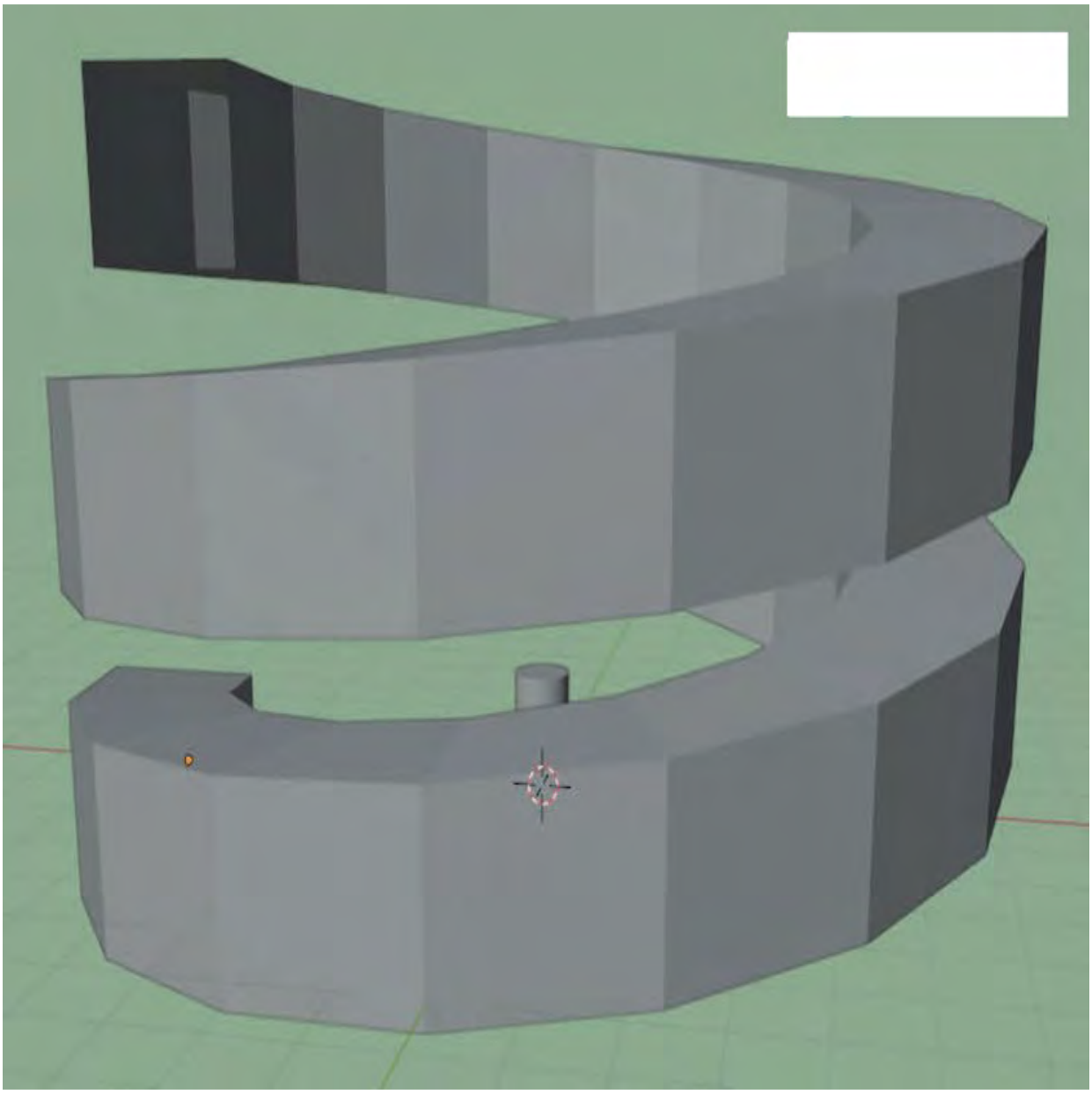


Figure 8.54

In Edit Mode you see the Vertices of the Plane duplicated in a circle about the vertical Axis (blue line) of the Cylinder Object (Figure 8.52).

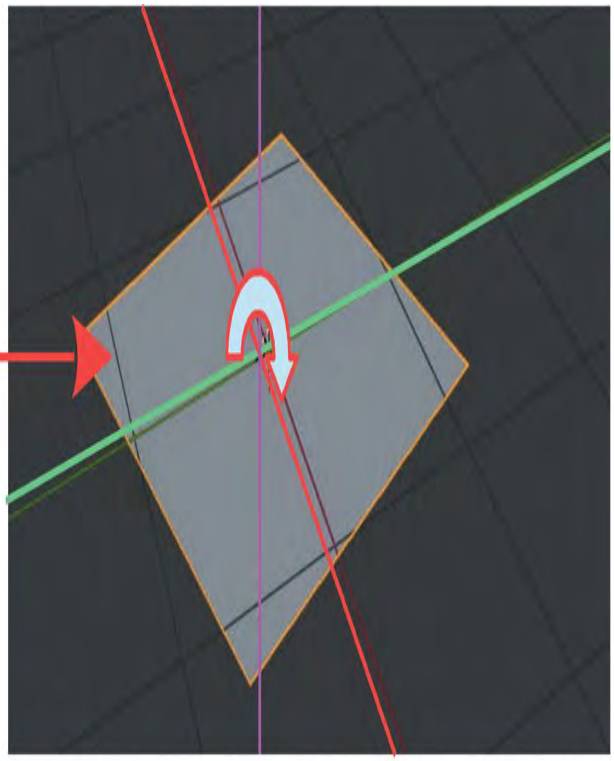
The circle is determined by the **Angle 360°** value in the Modifier panel.

By increasing the Screw value the last duplication is displaced vertically forming a Screw effect (Figure 8.53).

8.13 Skin Modifier

The **Skin Modifier** allows you to create a three dimensional shape from a basic stick arrangement consisting of a minimal number of Vertices. To demonstrate how this is achieved have a Plane Object in the 3D Viewport Editor. The Plane is a simple Object with four Vertices, four Edges and one Face.

Rotate the Plane about the Z Axis. This simply moves the Edges of the Plane away from the background grid lines to improve visibility when in Edit Mode.


Background Grid Line

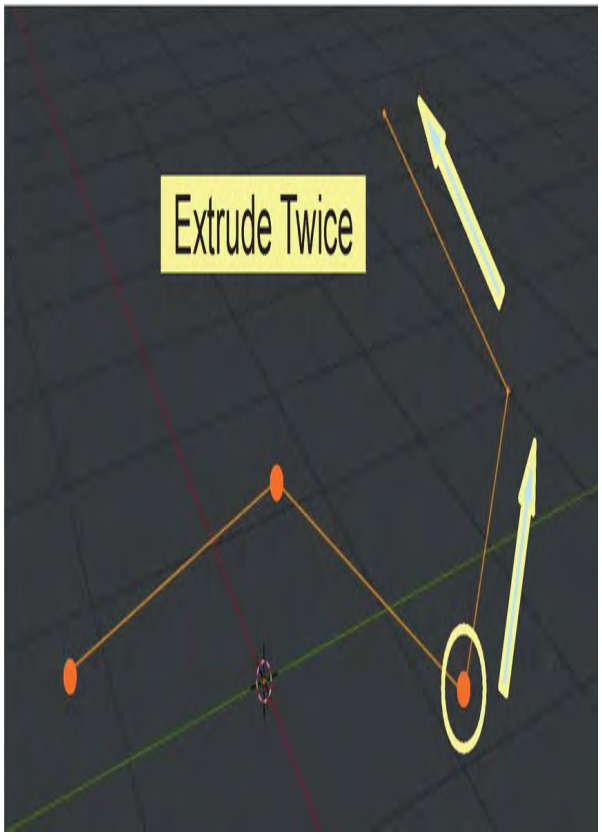

Extrude Twice

Figure 8.55

In **Edit Mode** delete one Vertex leaving two Edges (the basic stick arrangement).

Select one Vertex and extrude twice. Remember; the original Edges are located on the X – Y Axis of the Scene. The two new Edges will be drawn in a plane passing through the center of the Scene normal to the computer Screen.

In Object Mode, with the stick figure selected, add a **Skin Modifier** (Figure 8.56) and **click Apply** to produce the solid shown in Figure 8.57.

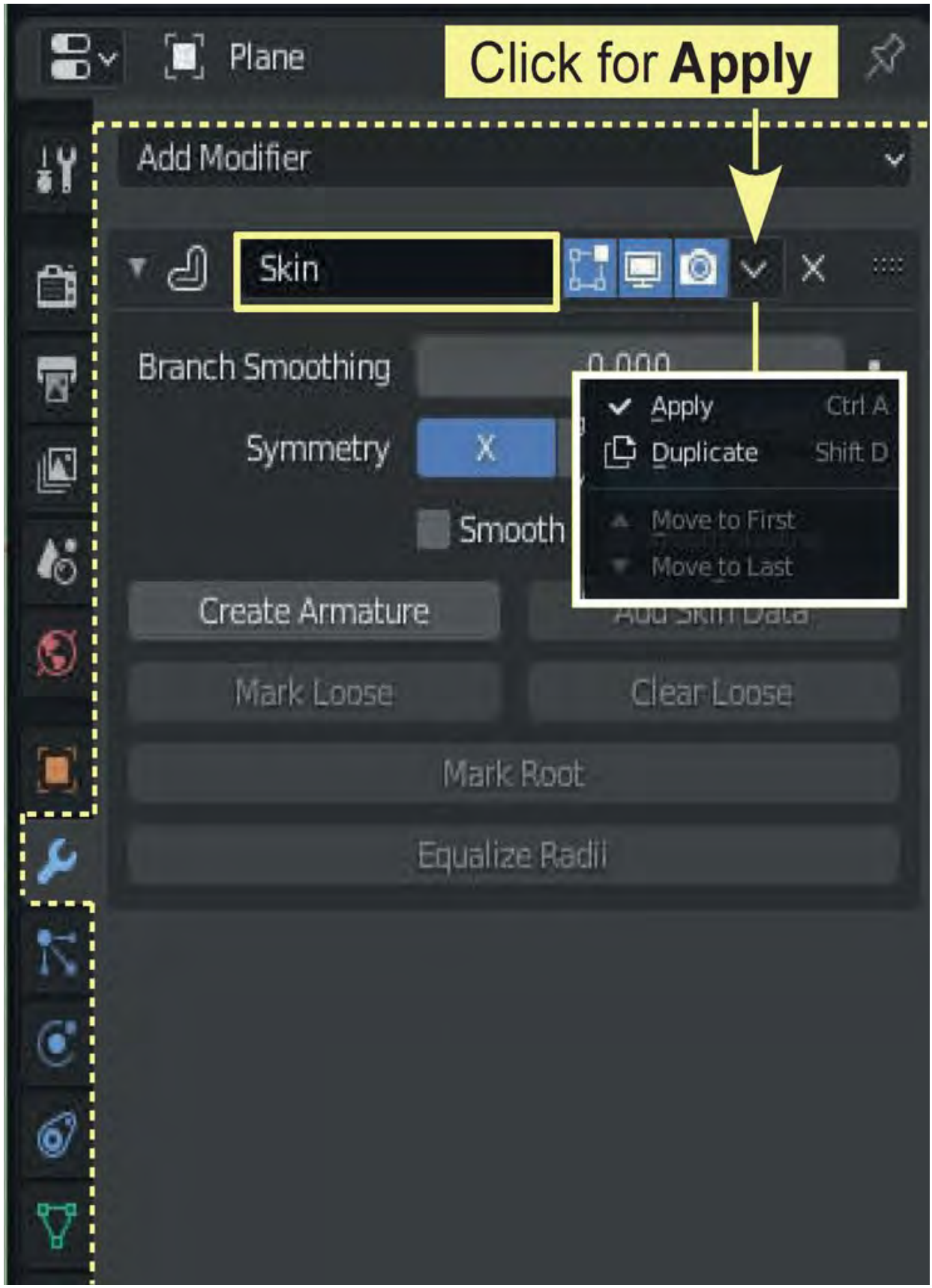

Plane
Click for Apply
Add Modifier
Skin
Branch Smoothing
Symmetry
X
Smooth
Apply Ctrl A
Duplicate Shift D
Move to First
Move to Last
Create Armature
Mark Loose
Clear Loose
Mark Root
Equalize Radii

Figure 8.56

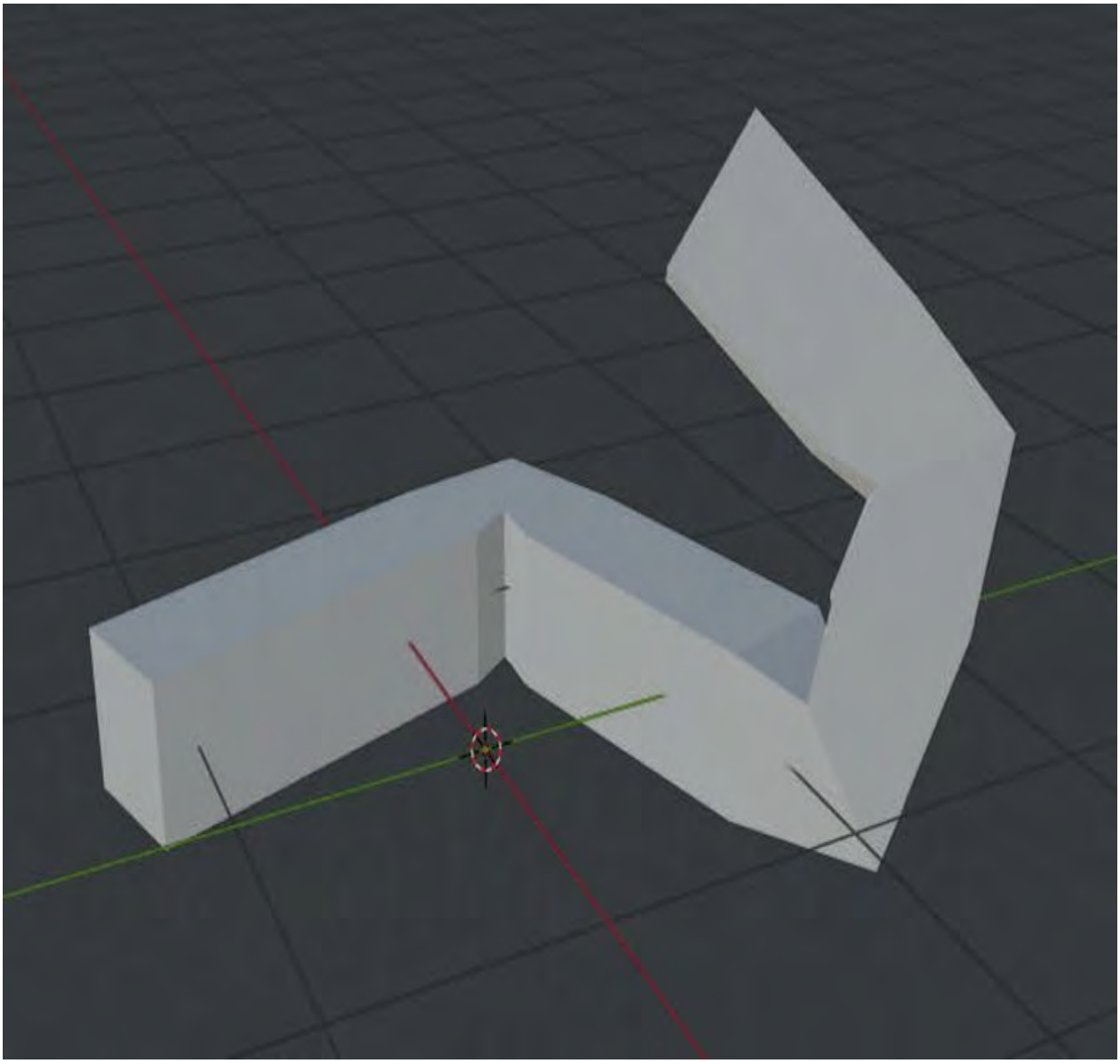
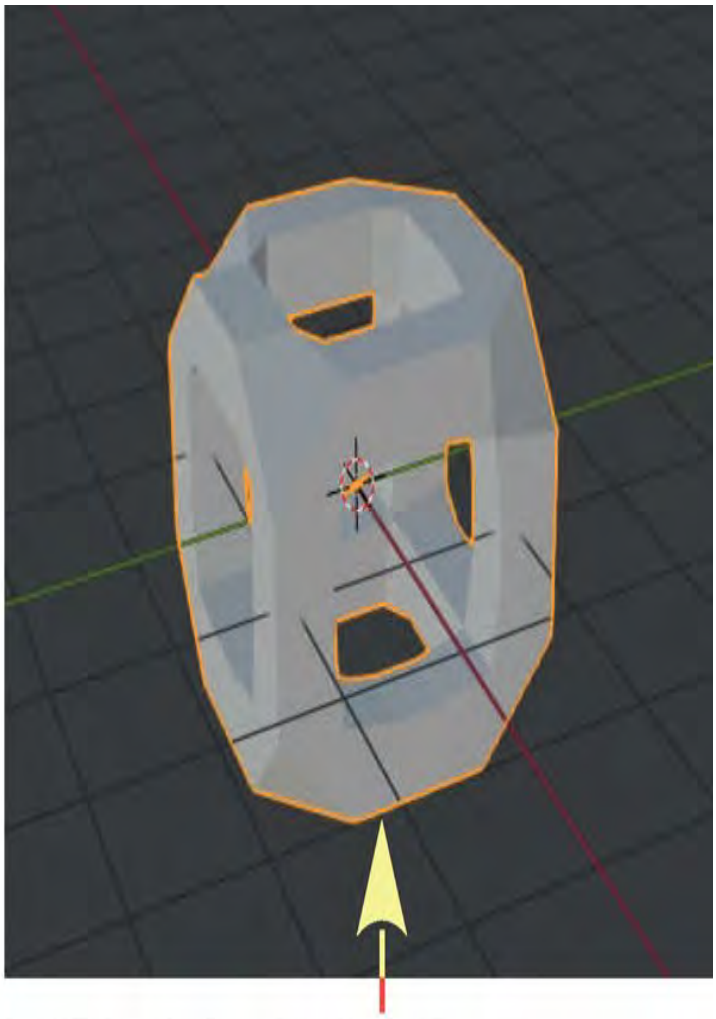


Figure 8.57

The Skin Modifier procedure, in essence, entails Extruding Vertices from a simple Object to form a rudimentary stick shape in Edit Mode. The Skin Modifier is then added to the shape in Object Mode. The Modifier creates a cage around the stick which converts to a Mesh when the Modifier is Applied.



Skin Modifier Applied to the default Cube

Figure 8.58

At this point, note the distinction between **Adding the Modifier** and **Applying the Modifier**. You add the Modifier in the Properties Editor, Modifier Properties by clicking the Add Modifier button and selecting from the menu. The Modifier panel displays where you adjust values to affect the selected Object in the 3D Viewport Editor. When adjustments are complete you Apply the Modifier by clicking the Apply button in the Modifier panel.

8.14 Solidify Modifier

The **Solidify Modifier** (Figure 8.59) provides a tool for creating solid Objects from thin-walled Objects. To demonstrate this, begin with a simple Plane Object selected in the 3D Viewport Editor. Add a **Solidify Modifier** in the **Properties Editor.**

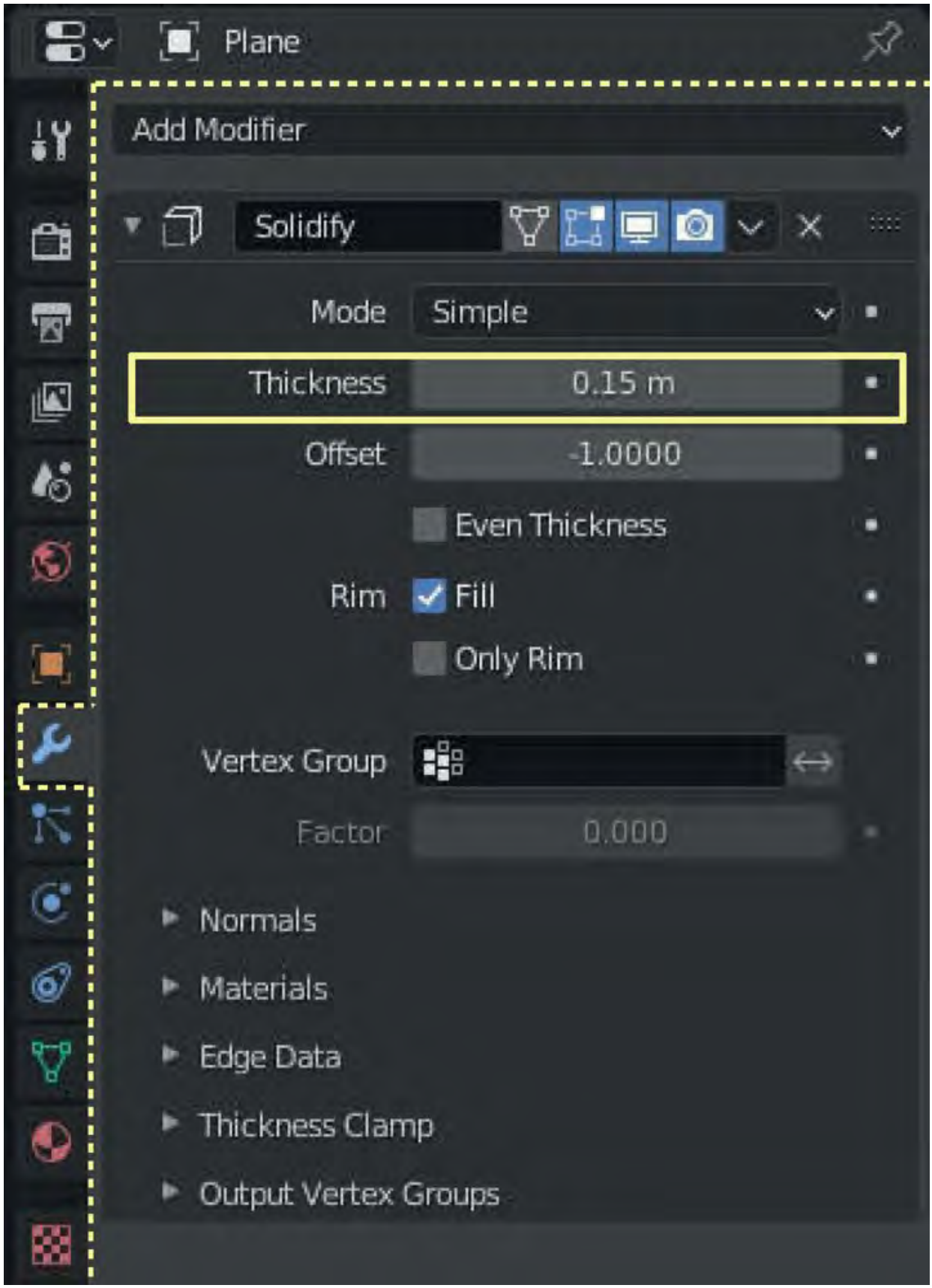

Plane
Add Modifier
Solidify
Mode
Simple
Thickness
0.15 m
Offset
-1.0000
Even Thickness
Rim
Fill
Only Rim
Vertex Group
Factor
0.000
Normals
Materials
Edge Data
Thickness Clamp
Output Vertex Groups

Figure 8.59

Look closely at the plane and you will observe that it now has a thickness (Figure 8.60).

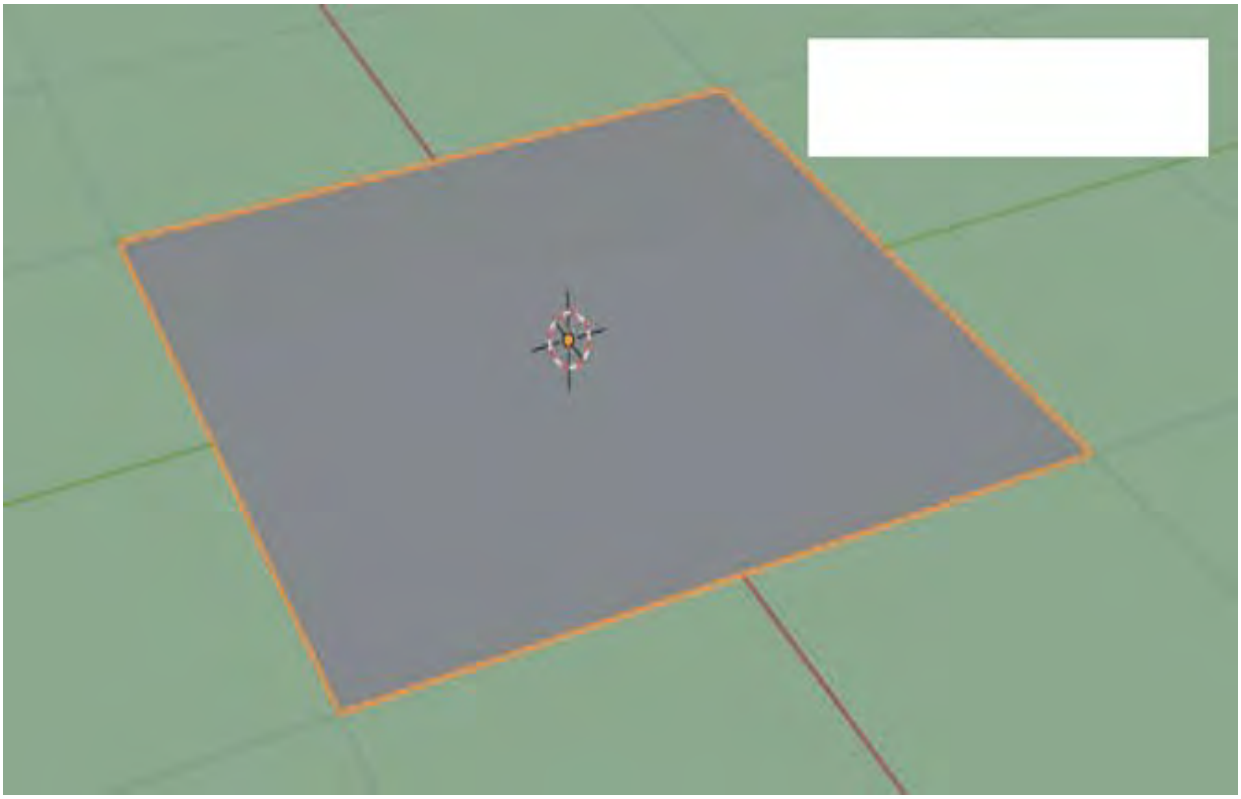


Figure 8.60

In the Modifier panel you will see **Thickness** 0.15m and **Offset** -1.0000.

Thickness is the thickness of the surface. Increasing this to 10 cm will give a better view (Figure 8.61).

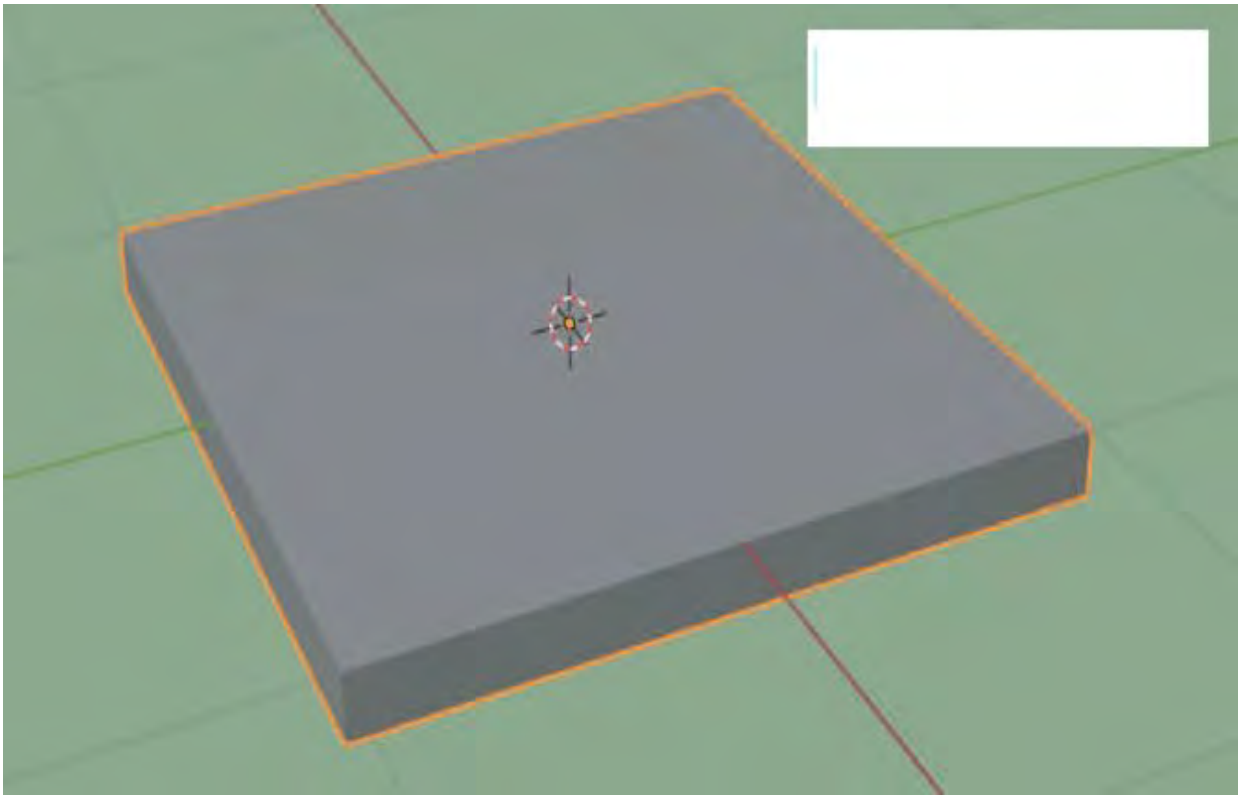
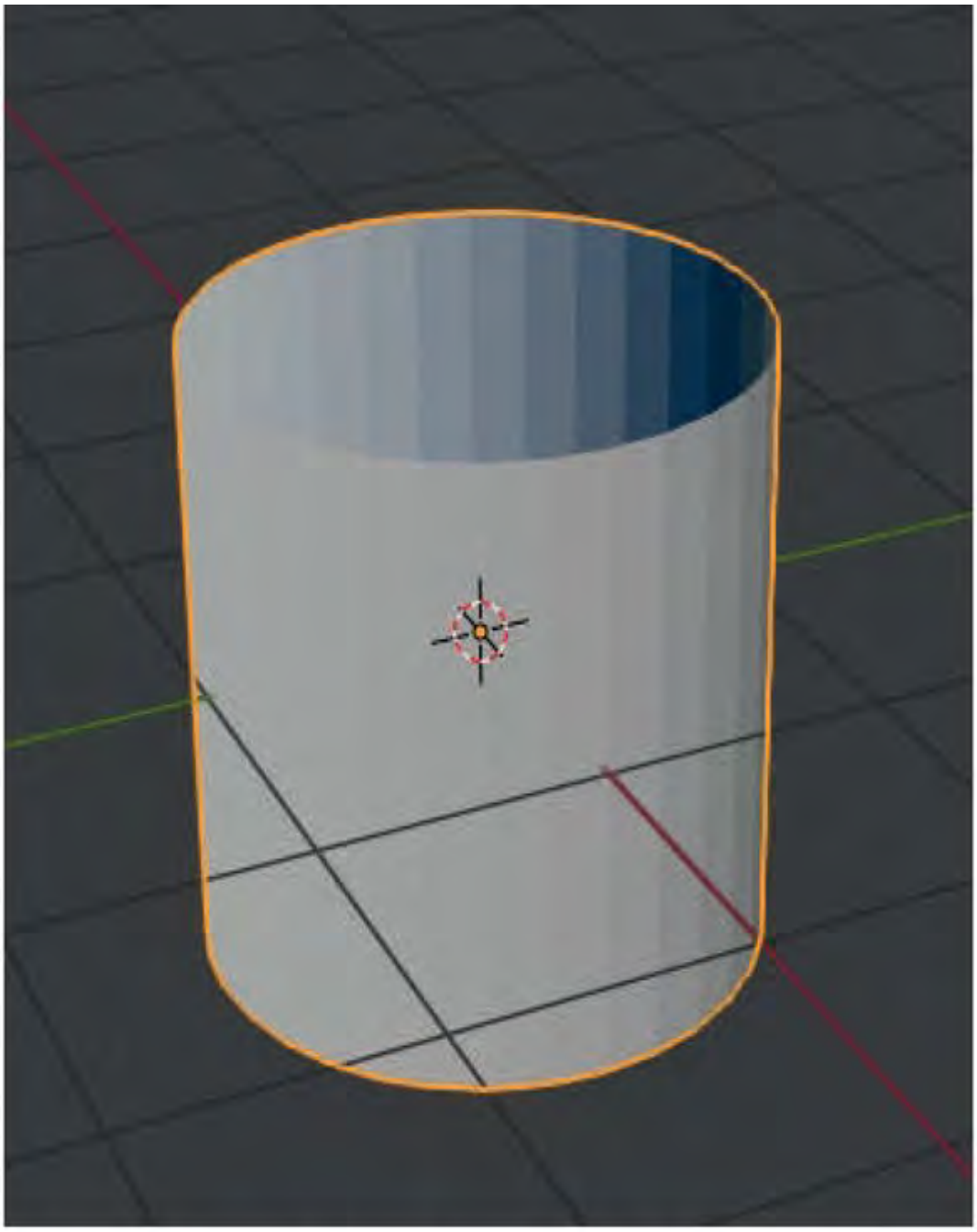


Figure 8.61

The Offset value range is -1.0000 to +1.000 which places the Thickness below or above the mid-plane. Offset 0.0000 has the Thickness straddling the mid-plane.

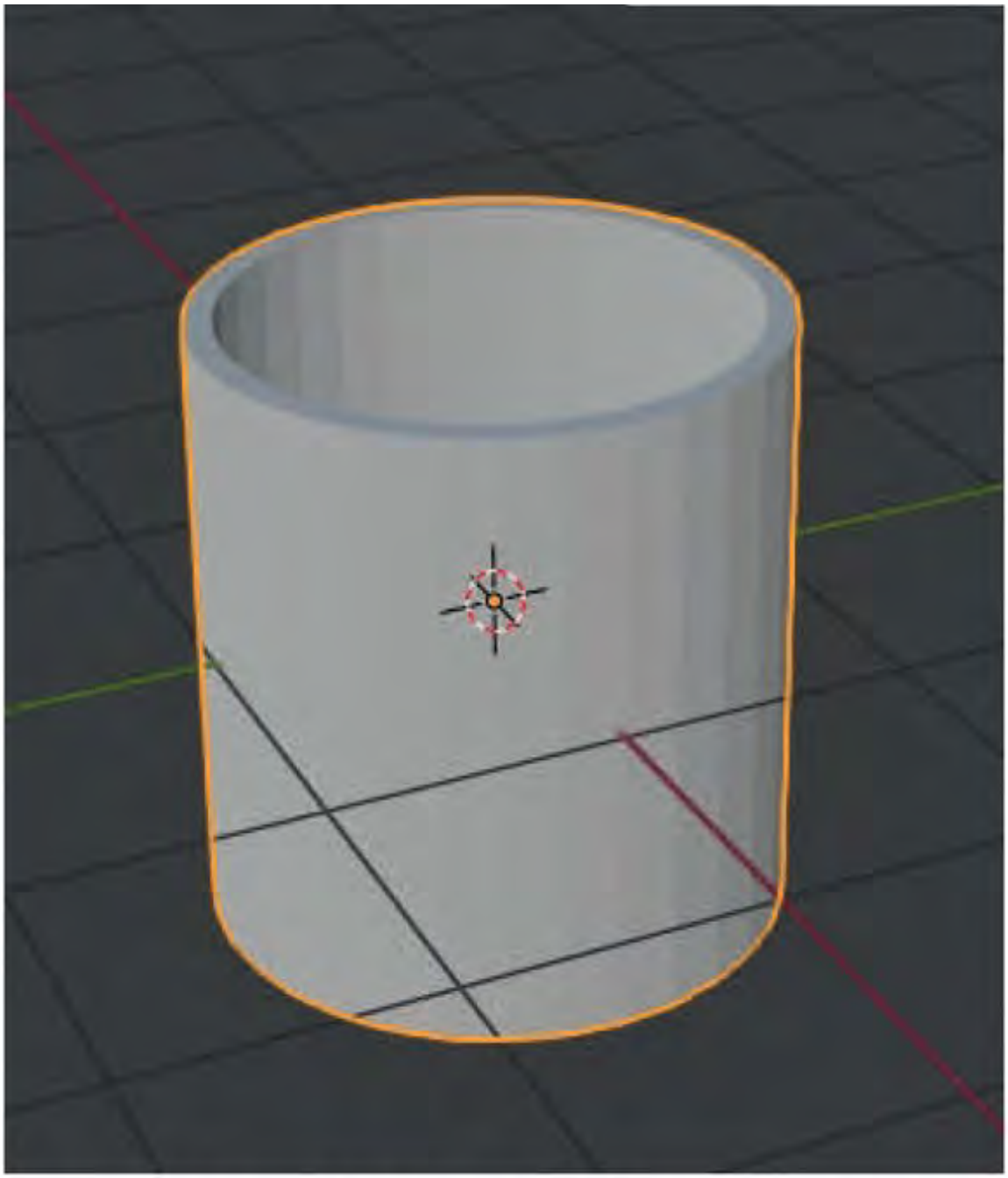
Tab to **Edit mode** and see that the original vertices of the plane object remain on the mid-plane of the Scene.

For a practical demonstration of how to use the **Solidify Modifier**, create a new Scene with a **Cylinder** object instead of the default Cube. **Delete the upper face of the Cylinder**. You now have a thin-walled container (Figure 8.62). In Object mode add a **Solidify Modifier** and change the thickness value 0.1m. The container will have wall thickness (Figure 8.63).



Original Cylinder – Top Face Deleted

Figure 8.62



Cylinder – Solidify Modifier Added Thickness 0.1m (10cm)

Figure 8.63

Note: Dimensions in Blender are proportional. A wall thickness of 0.01m (1.0cm) gives a thin wall thickness considering that the default overall dimensions of the Cylinder are 2m in diameter by 2m high (press the N Key to see Dimensions).

With Offset -1.0000 the wall thickness is created inside the original surface of the Cylinder. Offset +1.0000 creates the thickness outside the original surface.

In Edit Mode you see the original Vertices, Edges and Faces are unchanged until you click Apply in the Modifier Panel with the Cylinder in the 3D View Editor in Object Mode

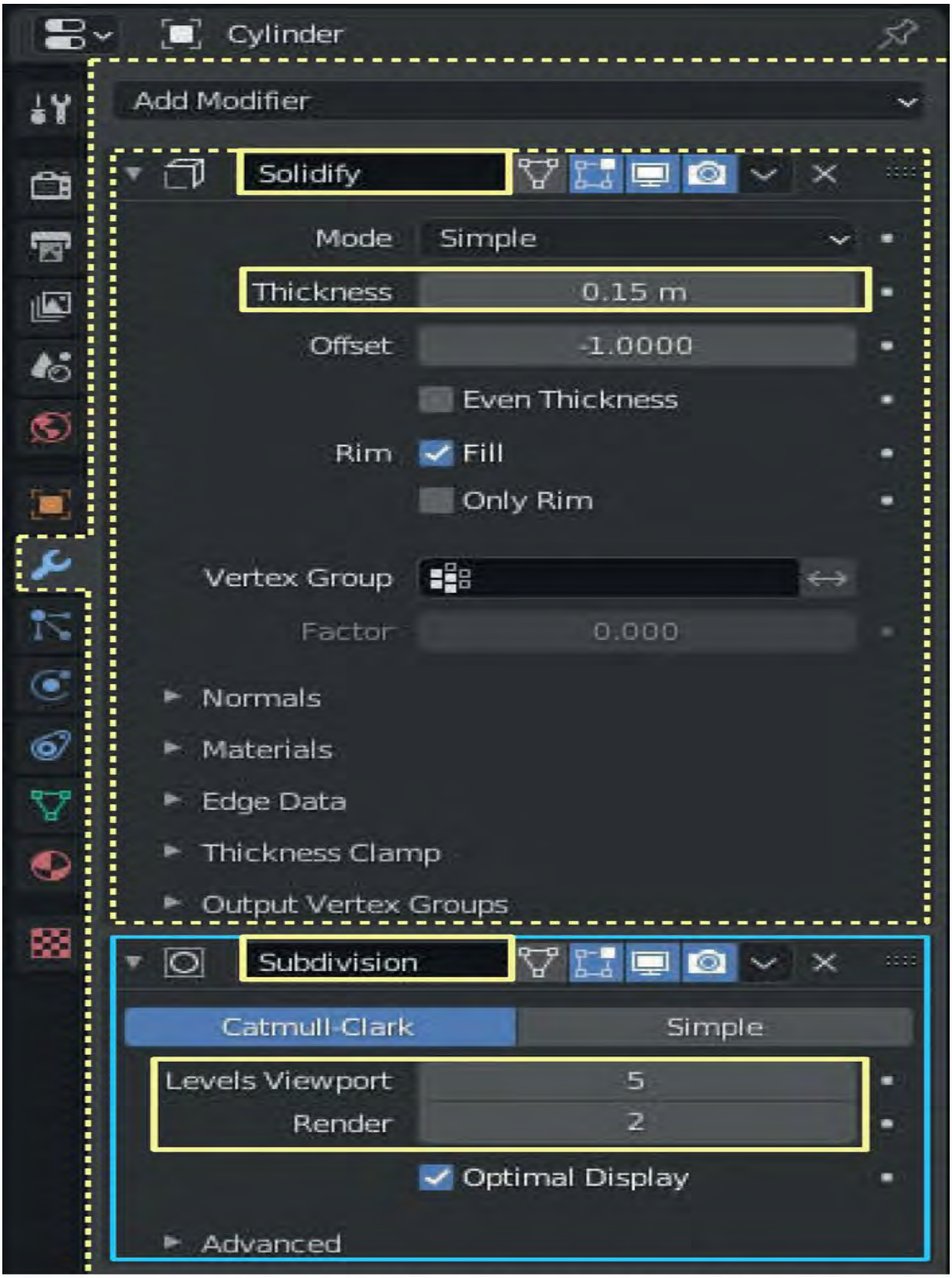
8.15 Subdivision Surface Modifier

The Subdivision Surface Modifier subdivides the surface of an Object adding Vertices, Edges and Faces giving the surface of the Object a smoother and rounded appearance. The additions also allow more detail to be modeled.

To demonstrate, use the container created in the previous exercise (Figure 8.63) with the Solidify Modifier Added but **NOT Applied,** add a **Subdivision Surface** (Subserf) Modifier.

When the Subdivision Surface Modifier is Added it is placed in the Modifier Stack in the Properties Editor below the Solidify Modifier. Remember; The Modifier at the top of the Modifier Stack takes precedence over Modifiers lower down. Therefore, the Solidify Modifier has precedence over the Subdivision Surface Modifier (Figure 8.64).

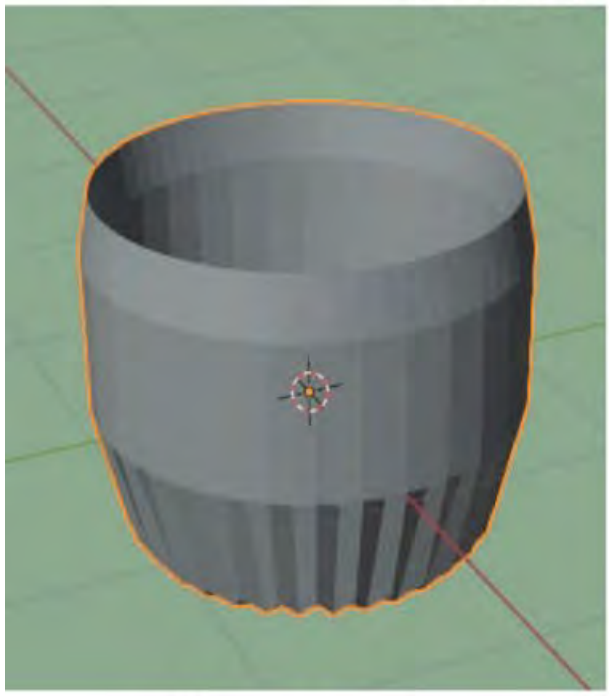
In the Subserf Modifier panel adjust the Subdivisions – **Level** values (Figure 8.64, 8.65).



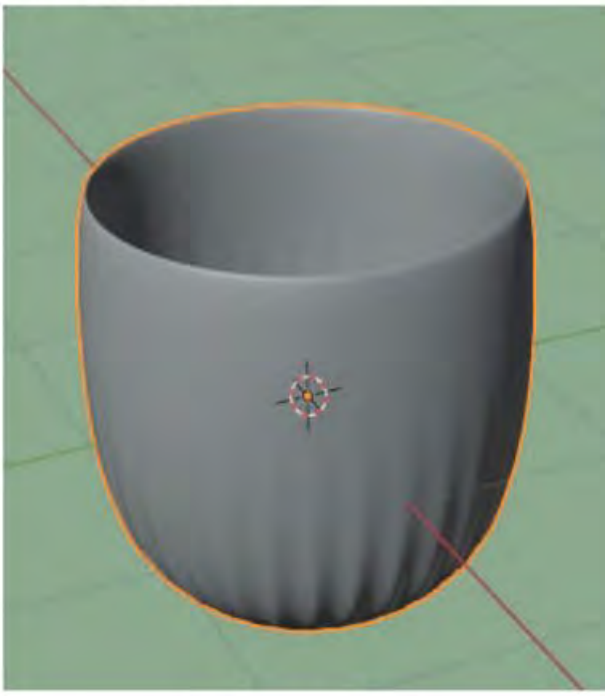
Blue outline indicates active Modifier

Figure 8.64

Subdivision – Levels



Viewport: 1



Viewport: 5

Render: 0

Render: 6

Figure 8.65

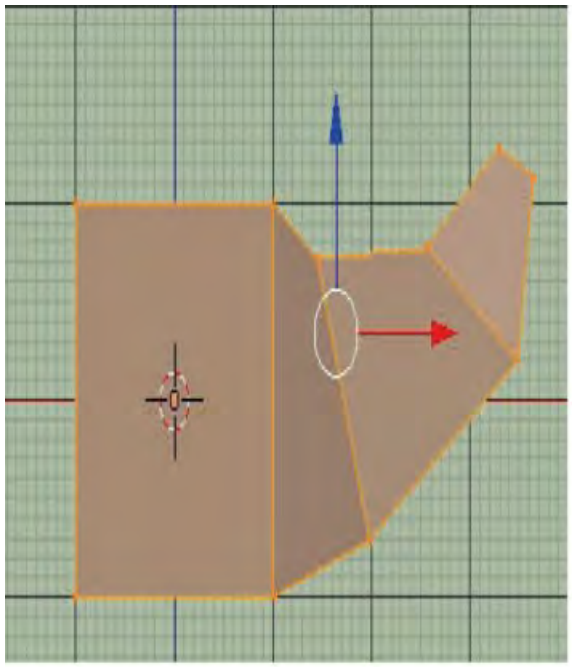
Note: Render Figures are Rendered Views (Press F12)

At this point neither the Solidify Modifier or the Subserf Modifier have been Applied. The Apply button, in both cases, has **NOT** been activated. The Modifiers display their effects in the 3D Viewport Editor but placing the Cylinder Object in Edit Mode shows that the original Vertices, Edges and Faces remain (Figure 8.66).

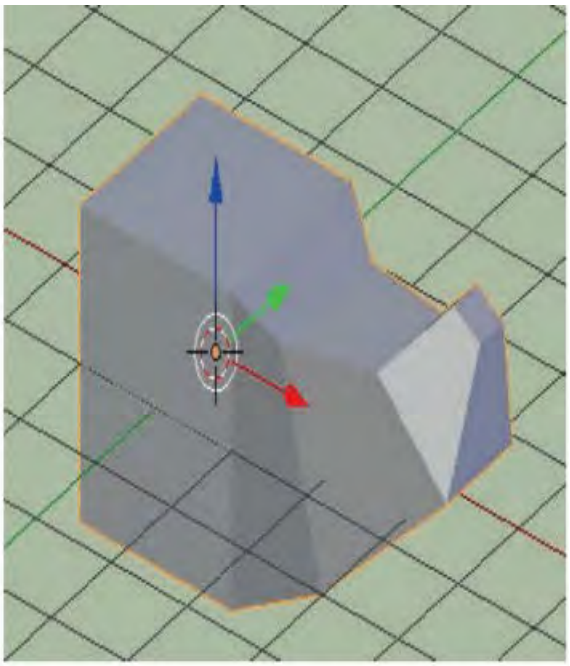
Figure 8.66

See over for another example of using the Subserf Modifier.

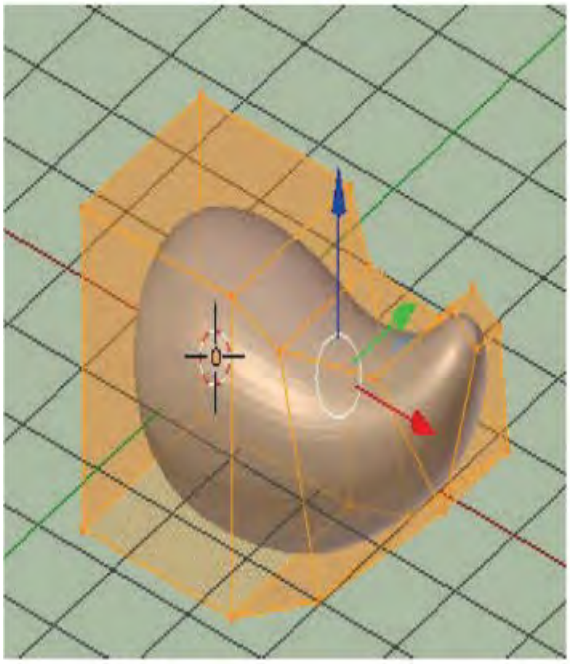
Another simple example using the **Subsurf Modifier** forms something like a whale's tooth (Figure 8.67).



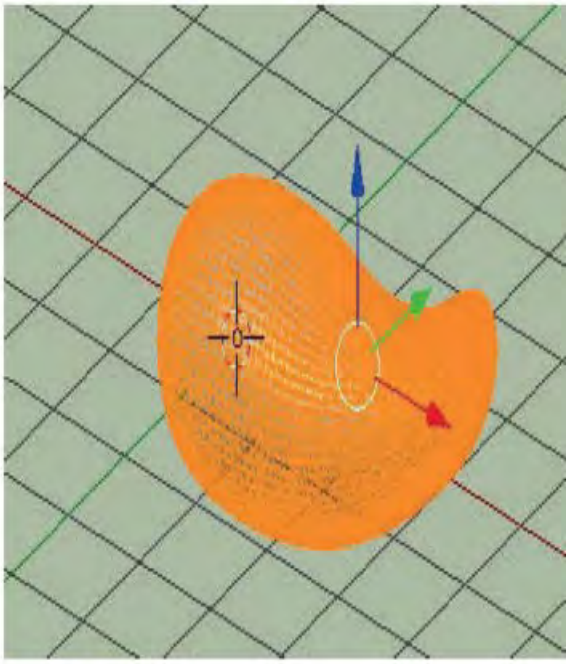
Cube Extruded – Edit Mode



Object Mode – 3D View Rotated



Object Mode – Subserf Added
Edit Mode – Show Original Mesh



Modifier Applied – New Mesh Created
Levels Viewport Value : 4

Figure 8.67

8.16 Triangulation and Weld Modifier

A **Triangulation Modifier** is added when a mesh model has been subdivided and Vertices added producing fine detail in a model. The Modifier ensures that triangulation will remain consistent when exporting or rendering.

If the model is animated using armatures the modifier should be placed in the modifier stack before (above) the armature modifier (Armatures: Chapter 22).

The Triangulation Modifier is also used when **Baking** prior to exporting and importing.

Baking is the process of precalculating and storing data to save time when converting what you have in the Viewport into an Image or Video file. The conversion is called **Rendering** (Chapter 19).

The **Weld Modifier** is used to merge Vertices in a non destructive manner. At this point the definition is probably meaningless, therefore a simple explanation is required.

In complex modeling, sometimes, Vertices will be duplicated, that is, two Vertices will be generated at the same point in the same 3D Space. It is advisable that the number of Vertices be kept at a minimum especially when Animation is involved and also to create the correct visual appearance.

Figure 8.68 To demonstrate, a shape will be generated from a Plane Primitive using Modifiers.

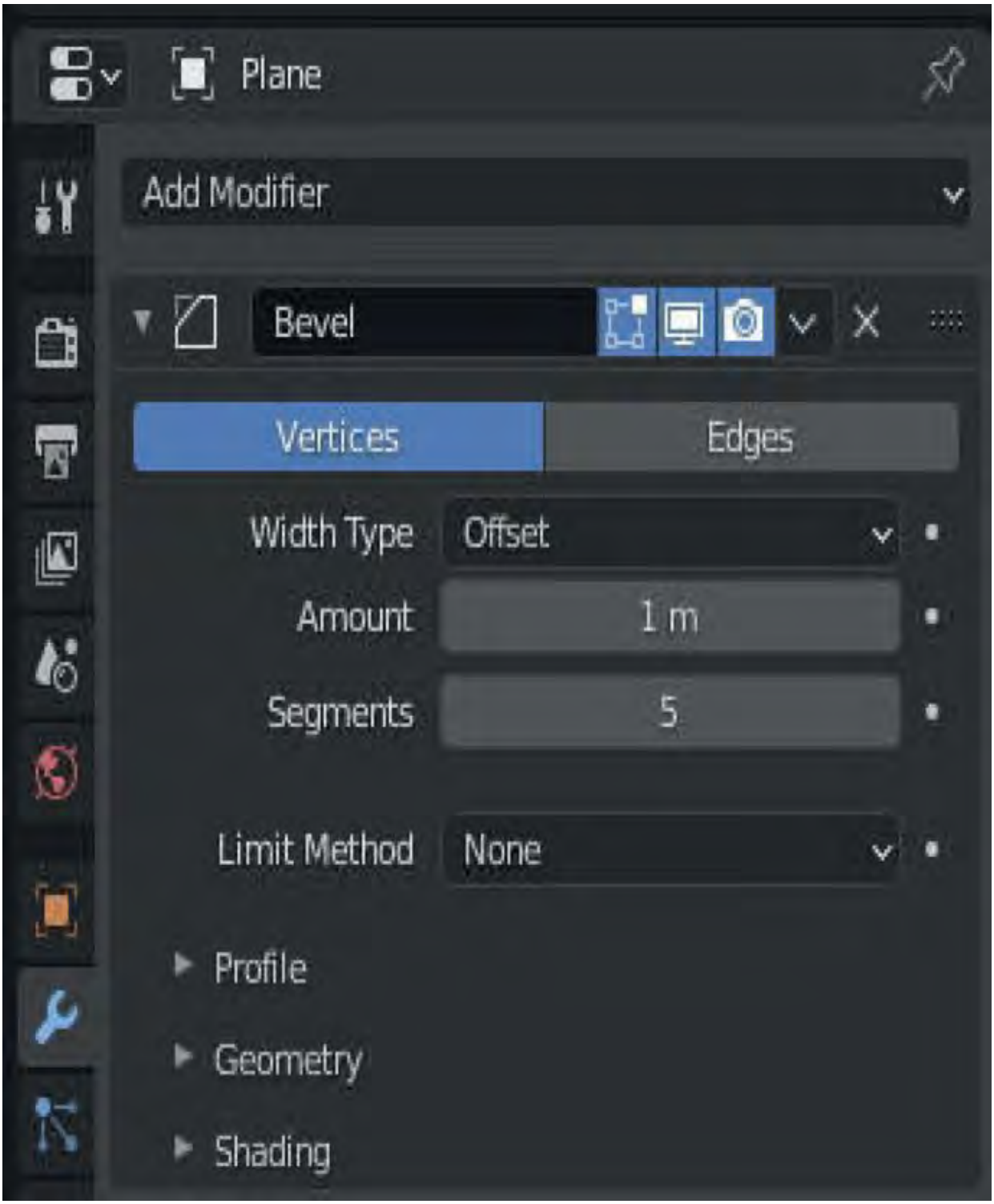


Figure 8.68

In the default 3D Viewport Editor delete the Cube Object and add a Plane. Scale the Plane three times on the X Axis and add a **Bevel Modifier** (Figure

8.73). (see 8.1 Modifiers – General – Bevel Modifier).

Important: While in Object Mode, press **Ctrl + A and select Apply – Scale** (Apply Object Transform to Scale).

In the **Properties Editor, Bevel Modifier**, check **Vertices**. Change the **Segments** to 5 then adjust the **Amount** value to 1 m to produce a complete rounded end on the Plane (Figure 8.69).

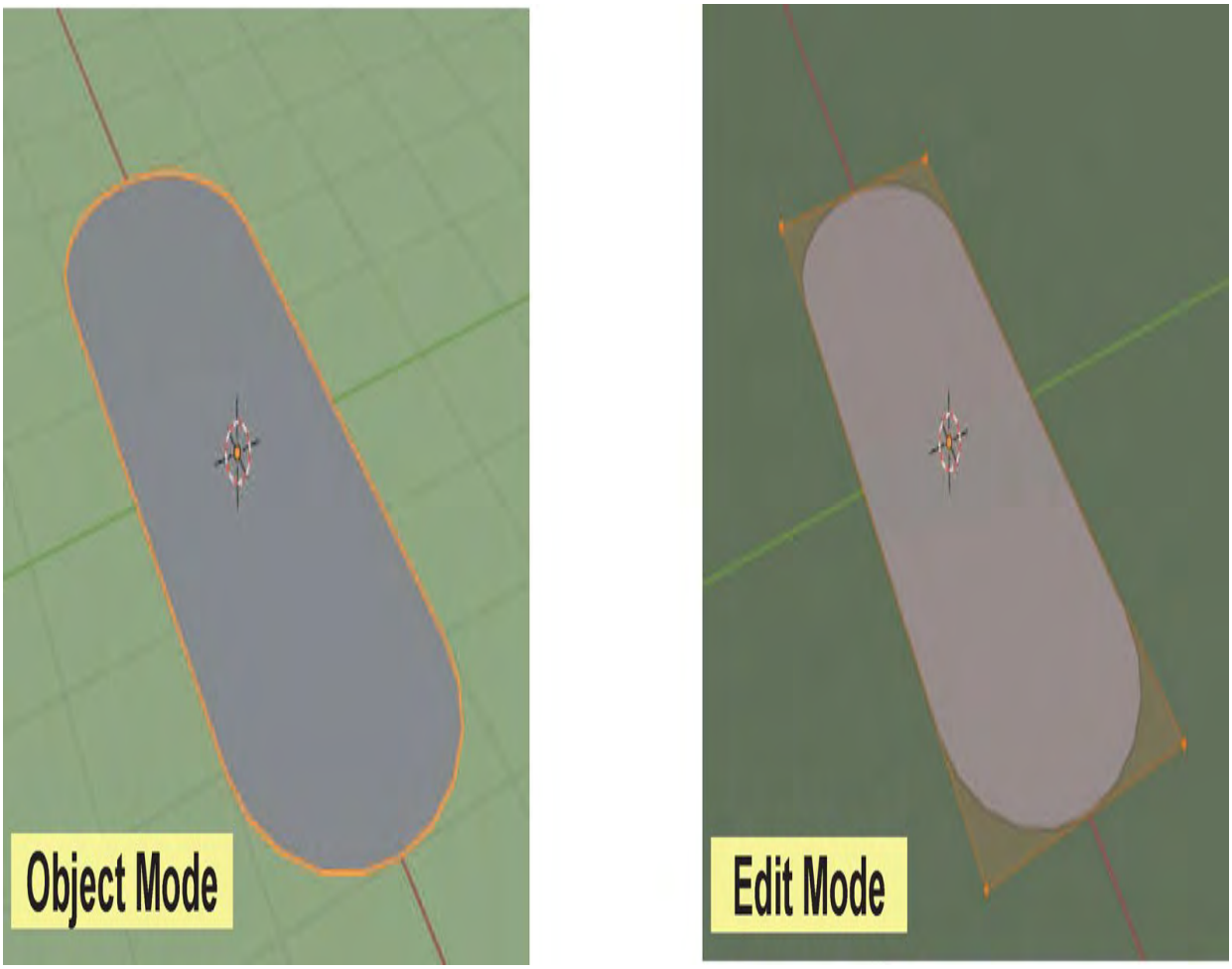


Figure 8.69

Solidify Modifier

Figure 8.70

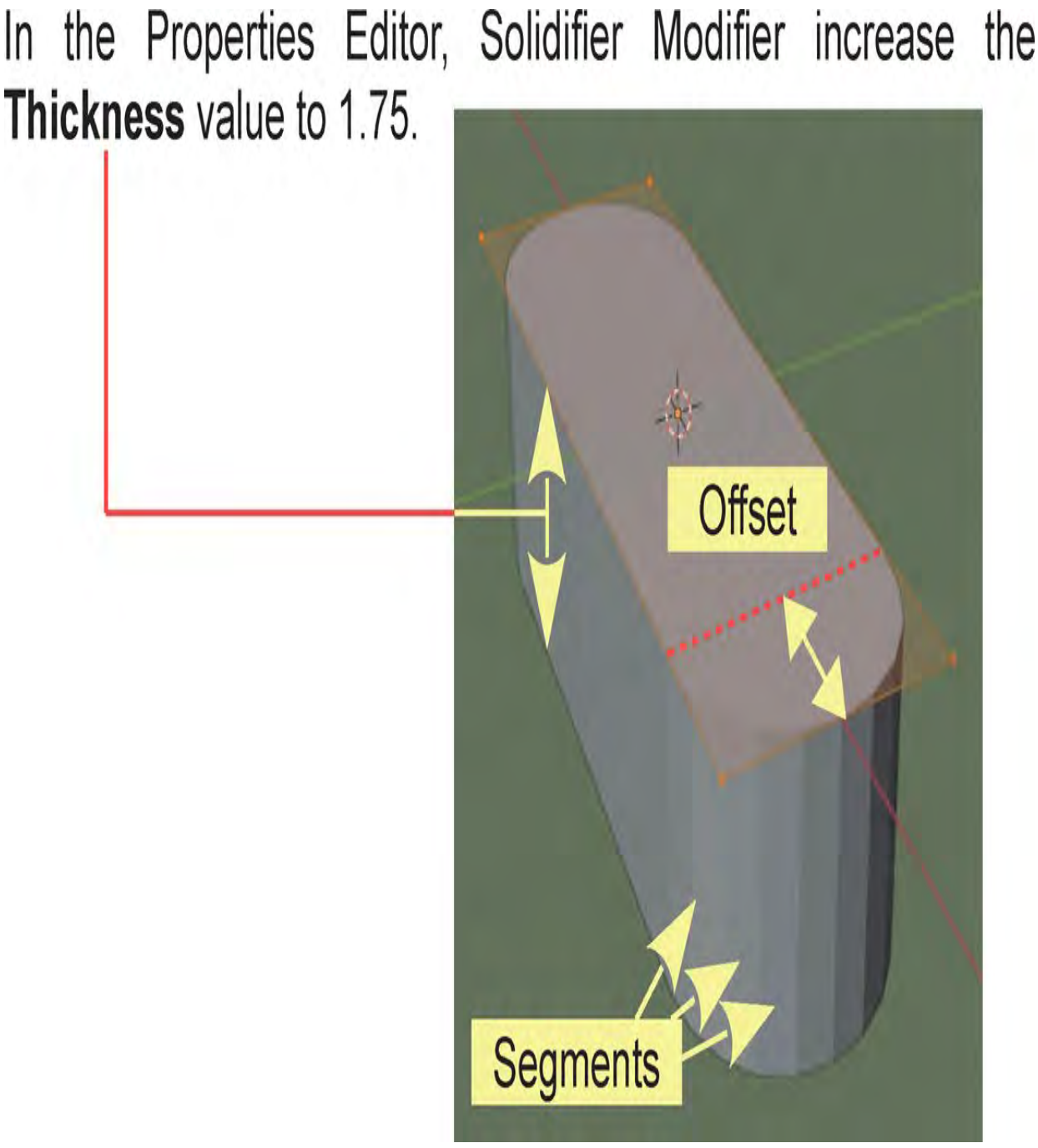


Figure 8.71

At this point, in Edit Mode you can see the results of the **Offset** value and **Segments** set in the Bevel Modifier.

In Object Mode RMB Click and select **Shade Smooth**.

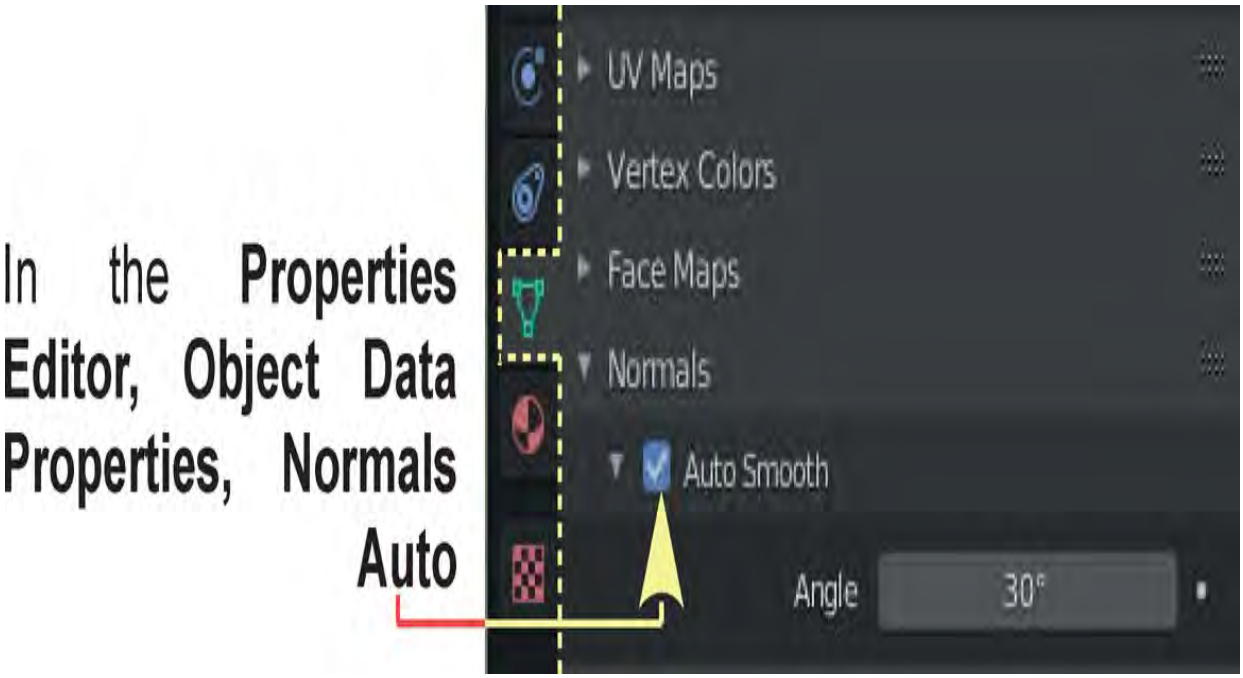


Figure 8.72

Figure 8.73

In Object Mode and Edit Mode you see the shape with round smooth ends. In Edit Mode you see that the original geometry (Four Vertices) is retained.

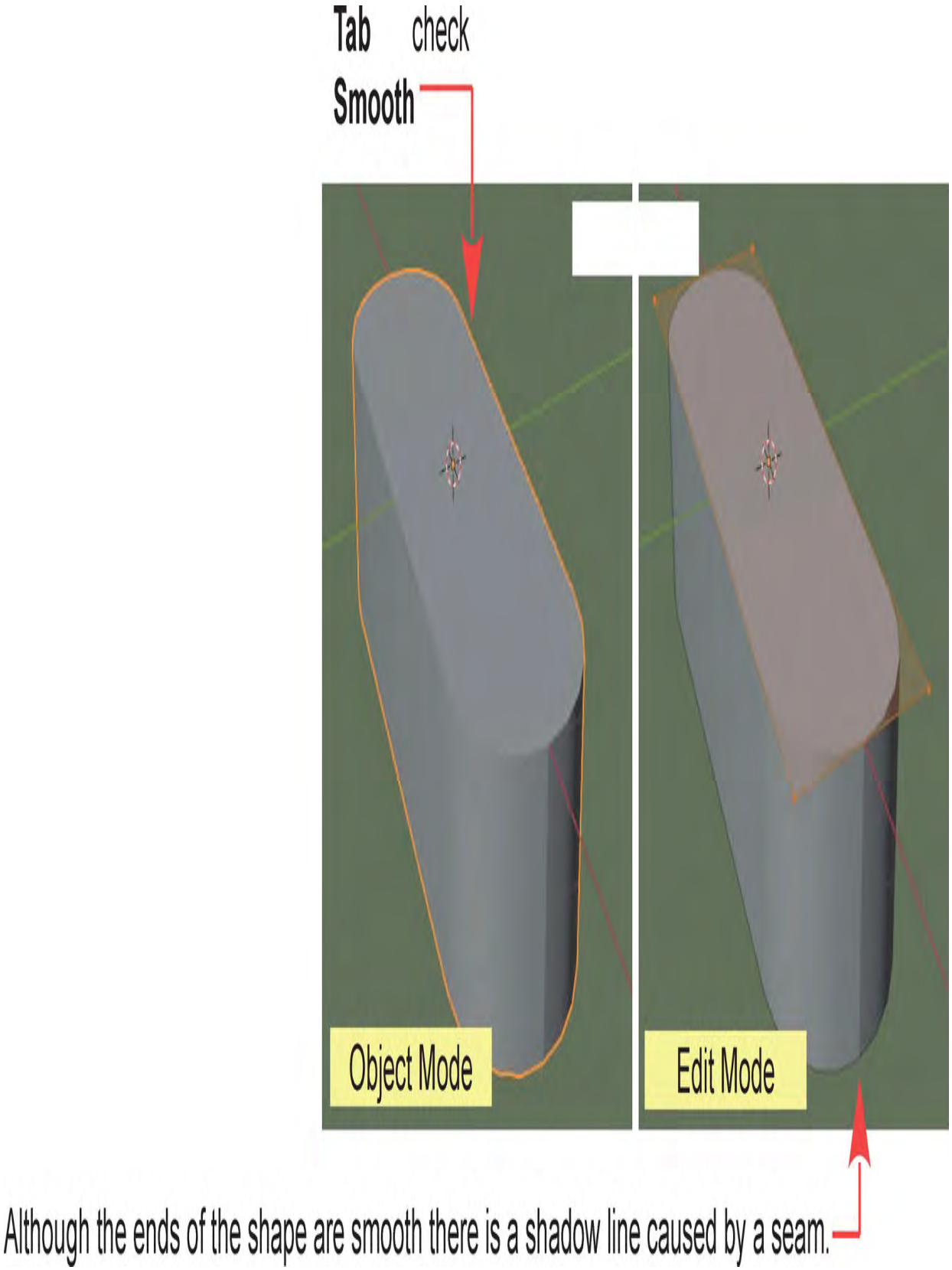

Tab check
Smooth
Object Mode
Edit Mode
Although the ends of the shape are smooth there is a shadow line caused by a seam.

Figure 8.74

Remember: The shape is generated using the Bevel Modifier. At this point the Modifier **has not** been applied as seen in Edit Mode but were the Modifier to be applied there would be a duplication of Vertices where the seam displays.

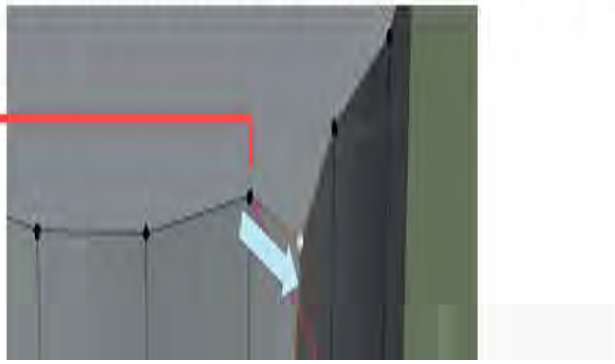


Figure 8.75

Since the Modifier is **NOT** applied you can **NOT** manually merge Vertices (remove the doubles).

Remember: In complicated models it is advisable not to apply the Modifiers to minimise the number of Vertices.

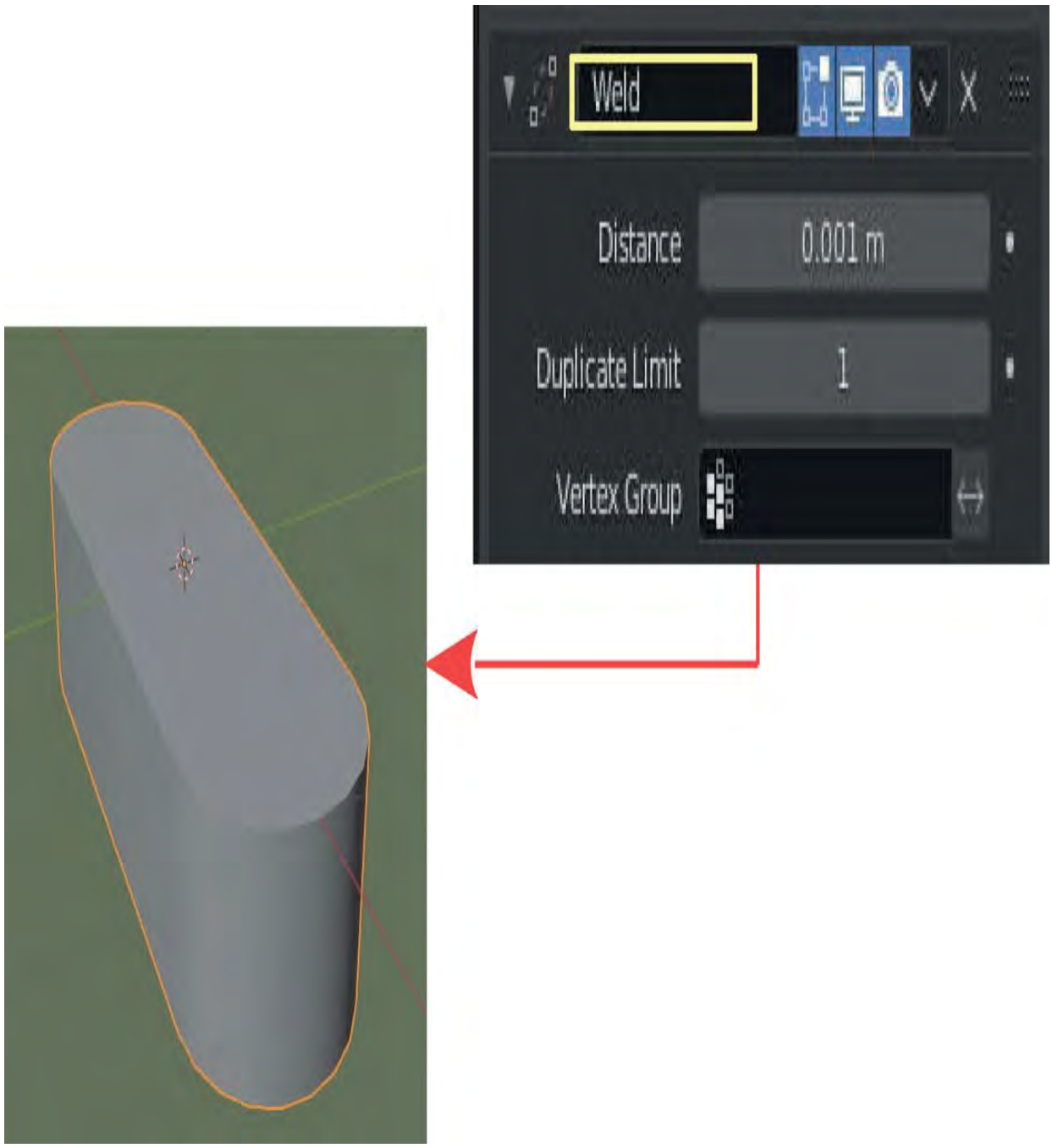


Figure 8.76
Figure 8.77

To clean up the seam add the **Weld Modifier**.

Note: Each Modifier Added has been placed at the bottom of the Modifier Stack.

8.17 Wireframe Modifier

The **Wireframe Modifier** converts a solid display as seen in **Solid Viewport Shading** to **Stick** or **Wireframe** display.

With the modifier added to a **Cube** object, instead of a solid Cube, you see a frame where the edges of the Cube have thickness (Figure 8.78).

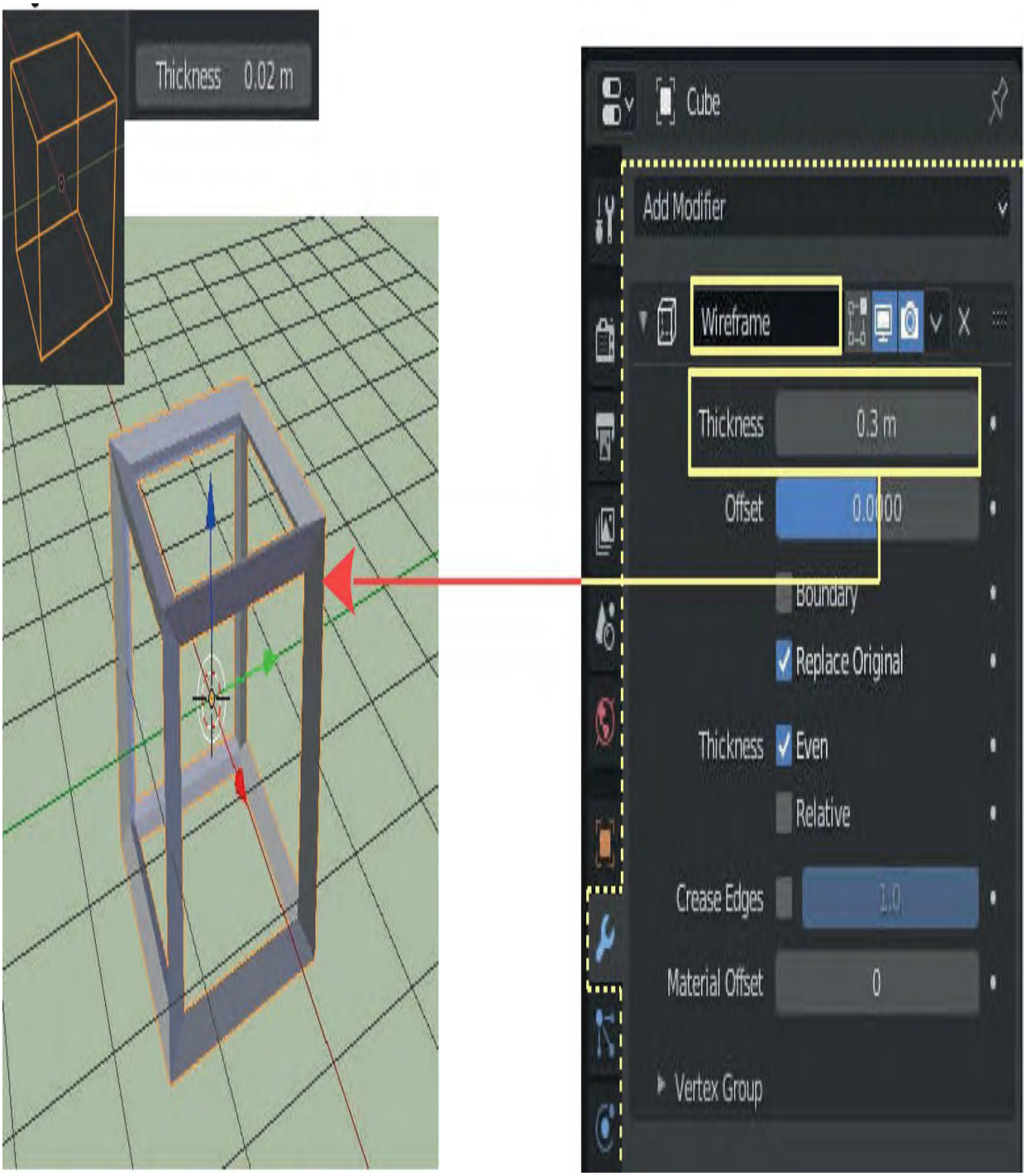


Figure 8.78

Figure 8.79 shows the Whale's Tooth from the previous example (Figure 8.67) with a Wireframe Modifier added.

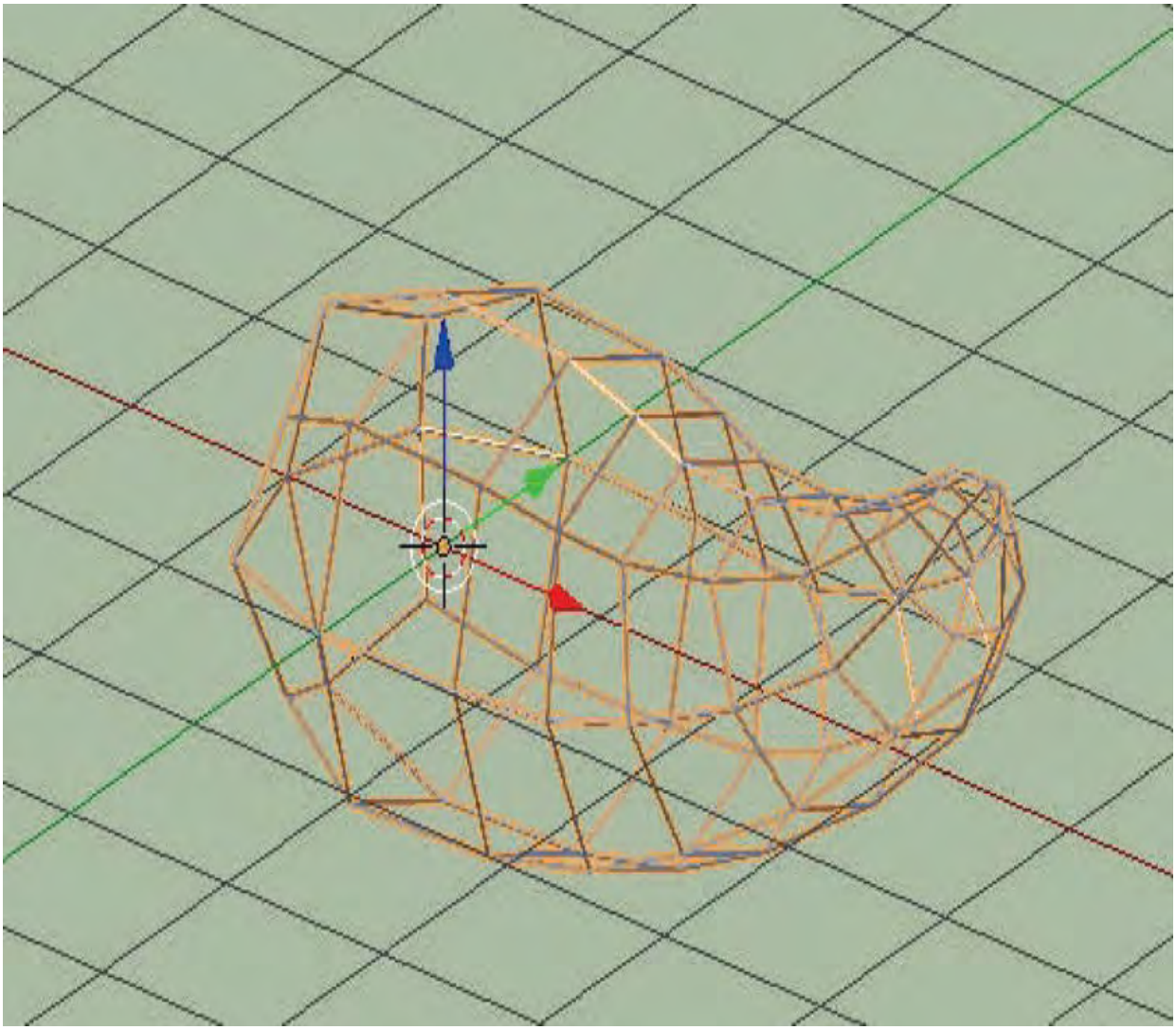
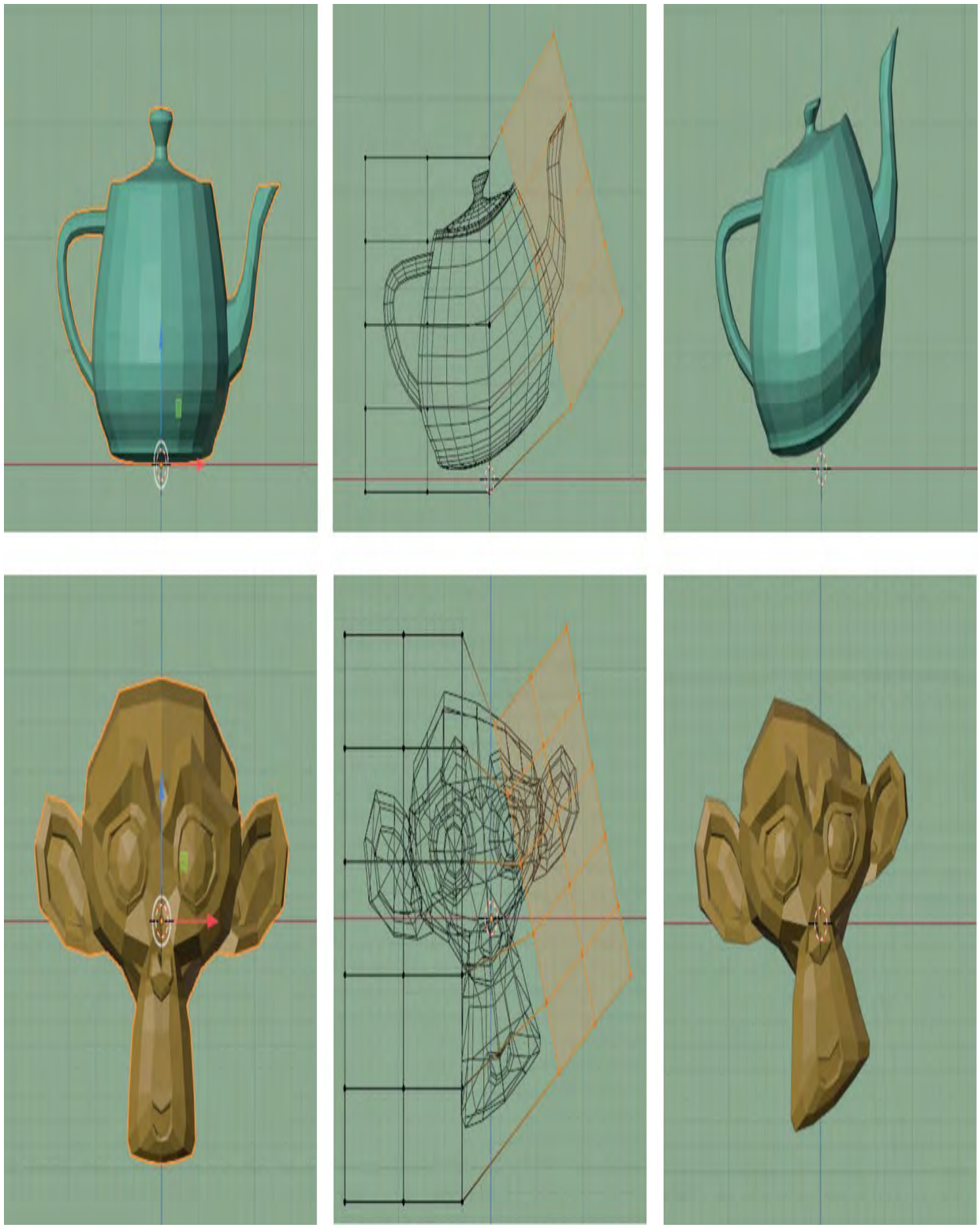


Figure 8.79

9 Editing with Deform Modifiers

9.1 Modifiers - Deform
9.2 Armature Modifier
9.3 Cast Modifier
9.4 Corrective Smooth Modifier
9.5 Curve Modifier
9.6 Displace Modifier
9.7 Hook Modifier
9.8 Laplacian Deform Modifier
9.9 Lattice Modifier
9.10 Mesh Deform Modifier
9.11 Shrinkwrap Modifier
9.12 Simple Deform Modifier
9.13 Smooth Modifier
9.14 Smooth Corrective Modifier
9.15 Smooth Laplacian Modifier
9.16 Surface Deform Modifier
9.17 Warp Modifier
9.18 Wave Modifier

The Deform group of Modifiers change the shape of an Object without adding new geometry, and are available for meshes, texts, curves, surfaces and/or lattices. The Mesh Deform modifier allows an arbitrary mesh (of any closed shape) to act as a deformation cage around a mesh Object.



9.1 Modifiers for - Deform

Modify
Data Transfer
Mesh Cache
Mesh Sequence Cache
Normal Edit
Weighted Normal
UV Project
UV Warp
Vertex Weight Edit
Vertex Weight Mix
Vertex Weight Proximity
Generate
Array
Bevel
Boolean
Build
Decimate
Edge Split
Mask
Mirror
Multiresolution
Remesh
Screw
Skin
Solidify
Subdivision Surface
Triangulate
Weld
Wireframe
Deform
Armature
Cast
Curve
Displace
Hook
Laplacian Deform
Lattice
Mesh Deform
Shrinkwrap
Simple Deform
Smooth
Smooth Corrective
Smooth Laplacian
Surface Deform
Warp
Wave
Simulate
Cloth
Collision
Dynamic Paint
Explode
Fluid
Ocean
Particle Instance
Particle System
Soft Body

Figure 9.1

The Deform group of Modifiers generally provide tools for modifying or deforming a mesh Object as a whole. Some of these modifiers require a knowledge of other Blender features and will, therefore, be described as those features are encountered.

9.2 Armature Modifier

Armatures in Blender are Objects used for manipulating and posing other Objects such as models of characters. Posing is the technique used when animating figures. The **Armature Modifier** is discussed in conjunction with Armatures and Character Rigging in Chapter 22.

9.3 Cast Modifier

The **Cast Modifier** is used to deform a primitive Object such as the Cube Object in the default Scene.

Select the Cube Object. In **Edit Mode,** Subdivide (number of cuts 4) (Chapter 5 — 5.5) then **Tab** back to **Object Mode.**

In the **Properties Editor, Modifier Properties** add a **Cast Modifier** (Figure 9.2). By changing the **Cast Type,** altering the **Factor, Radius** and **Size** values and or limiting the effects to the X,Y and Z axis the deformation of the cube is controlled.

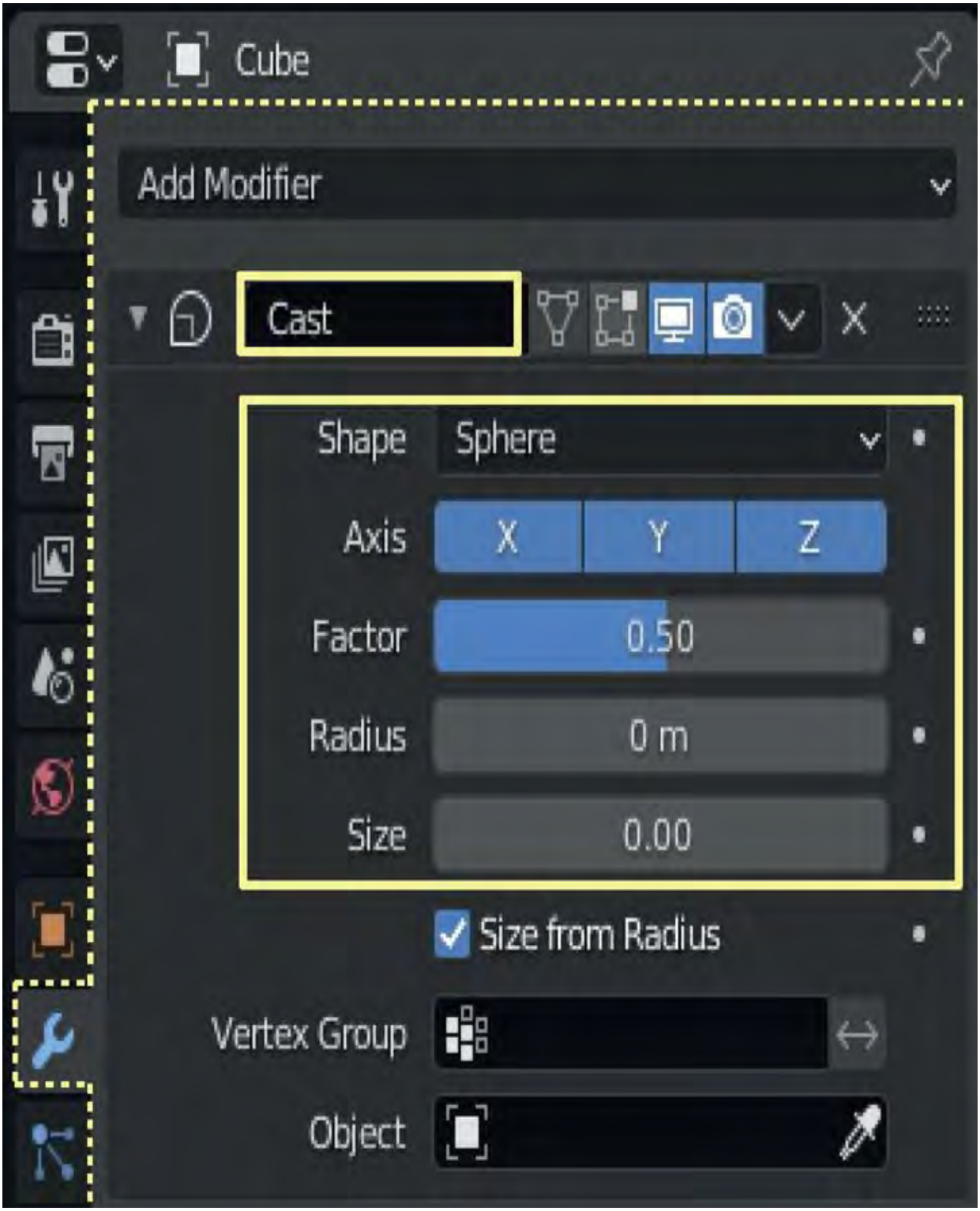

Cube
Add Modifier
Cast
Shape
Sphere
Axis
X
Y
Z
Factor
0.50
Radius
0 m
Size
0.00
Size from Radius
Vertex Group
Object

Figure 9.2

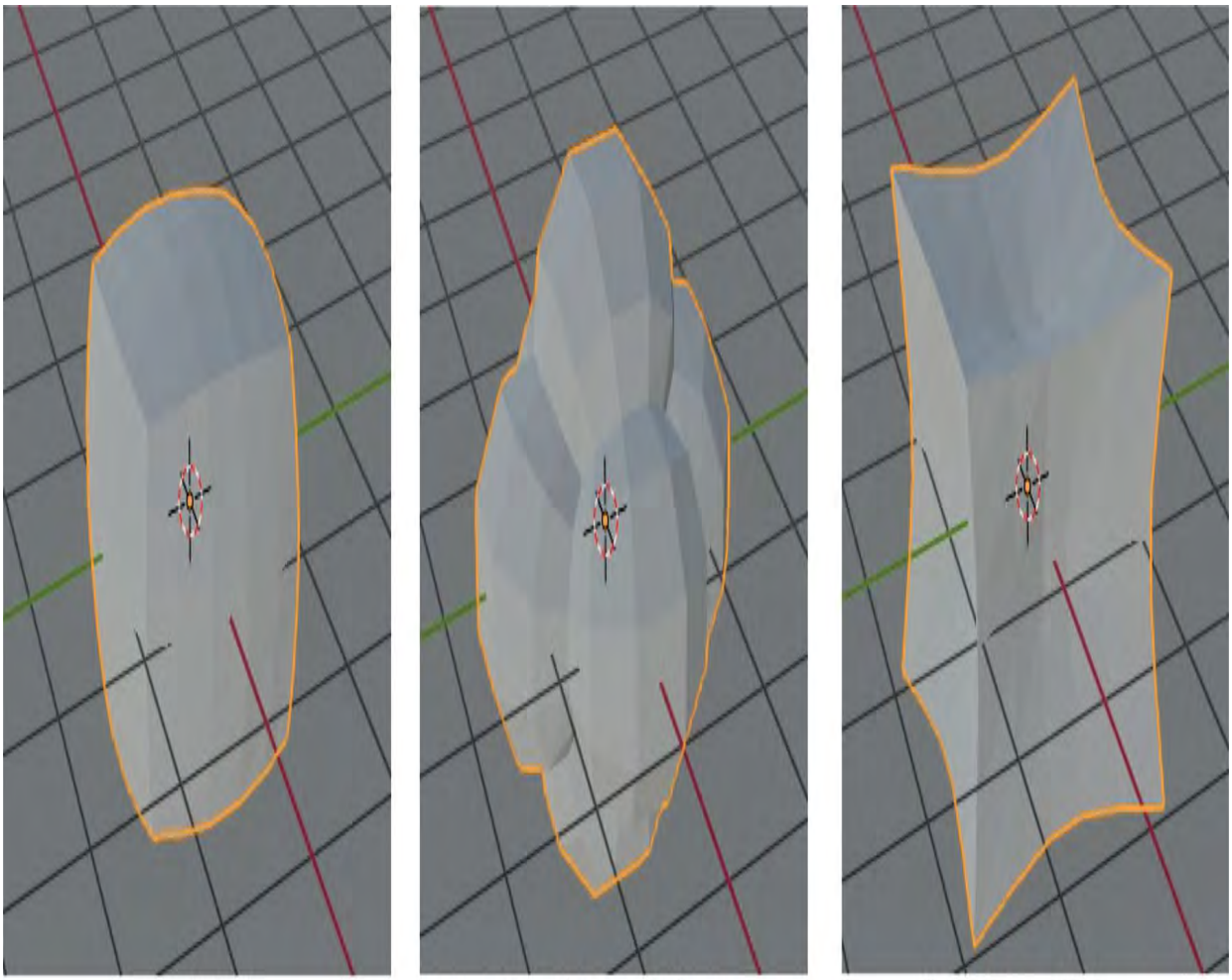

Default Cube
Subdivided Four Times
Cast Modifier Added
Type Sphere Factor: 0.50
Cast Type: Sphere
Factor: 2.000
Cast Type: Sphere
Factor: -2.000

Figure 9.3

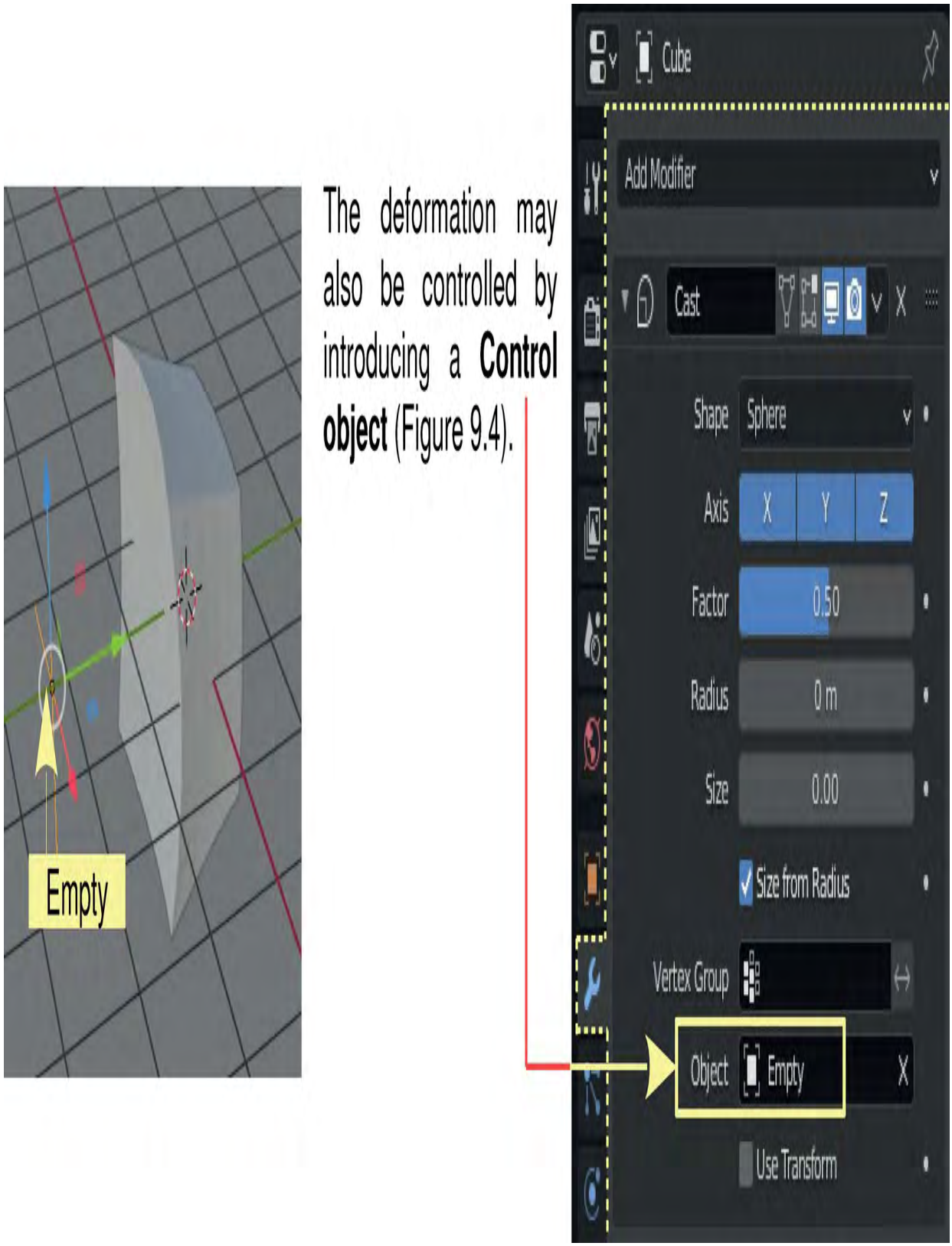

Empty
The deformation may also be controlled by introducing a **Control object** (Figure 9.4).
Cube
Add Modifier
Cast
Shape
Sphere
Axis
X
Y
Z
Factor
0.50
Radius
0 m
Size
0.00
Size from Radius
Vertex Group
Object
Empty
Use Transform

Figure 9.4

With the default Cube subdivided, deselect the Cube and place an **Empty Object** in the Scene. Add a **Cast Modifier** to the **Cube** and then click on the **Control Object** (cube icon) and select **Empty** from the menu to enter the **Empty** as the **Control Object.**

Move the Empty in the Scene to see the Cube being deformed.

Experiment with the Cast Type, the X,Y and Z axis settings and Factor, Radius and Size values. Keep in mind that the Empty Object may be animated to move in the Scene thus animating the deformation of the Cube.

9.4 Corrective Smooth Modifier

The **Corrective Smooth Modifier** is primarily designed to smooth incorrect mesh deformation which can occur when Armatures are used to deform a mesh.

9.5 Curve Modifier

The **Curve Modifier** uses the shape of a Curve to deform a mesh. Figure 9.5 shows the default Blender Scene with the Cube Scaled down, then Scaled along the Y Axis (S Key + Y Key) and Subdivided in Edit Mode with Number of Cuts = 10.

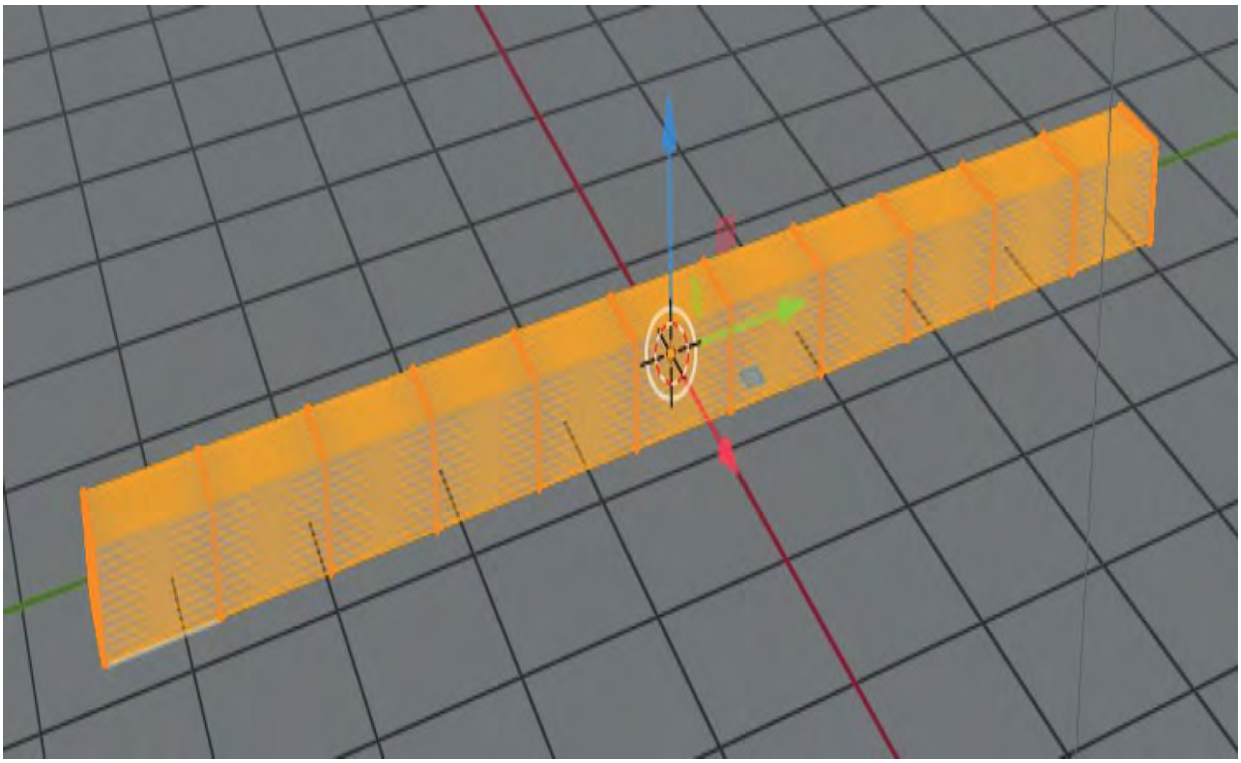


Figure 9.5

In **Object Mode** the elongated Cube is deselected and a **Bezier Curve** added (Chapter 10 — 10.2). The **Curve** is rotated and scaled up to match the length of the Cube (Figure 9.6).

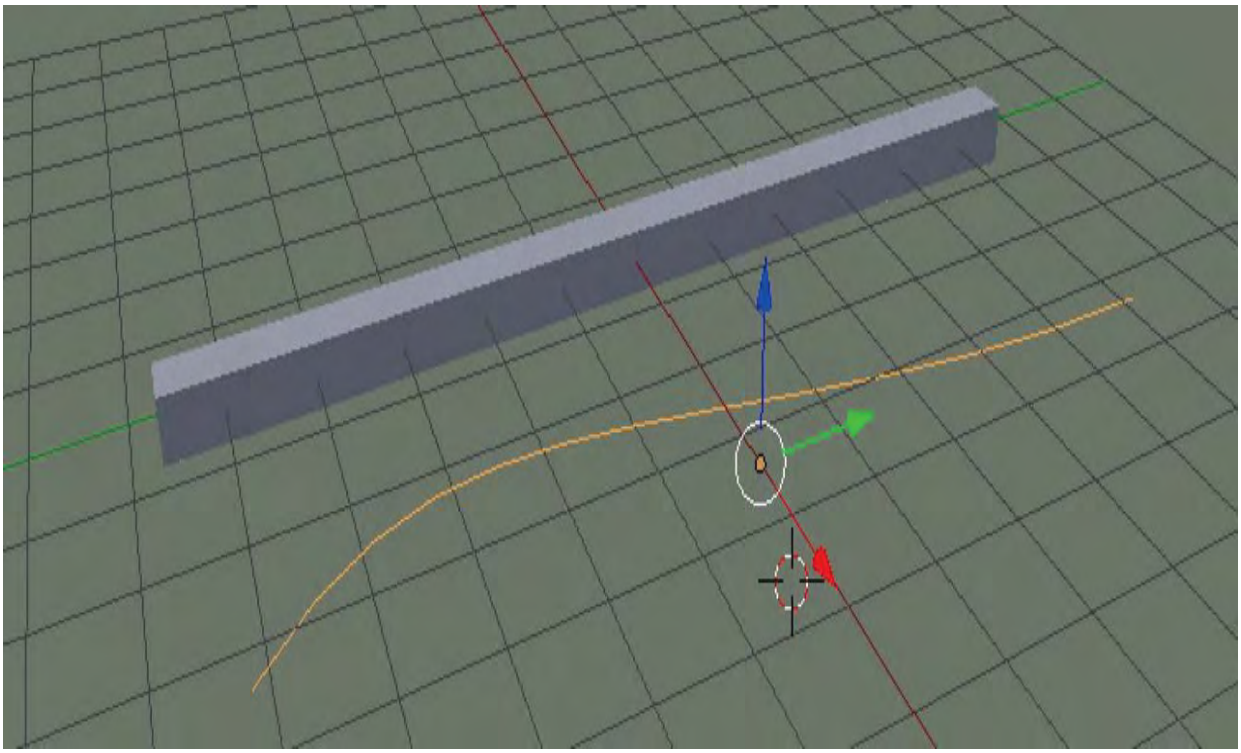
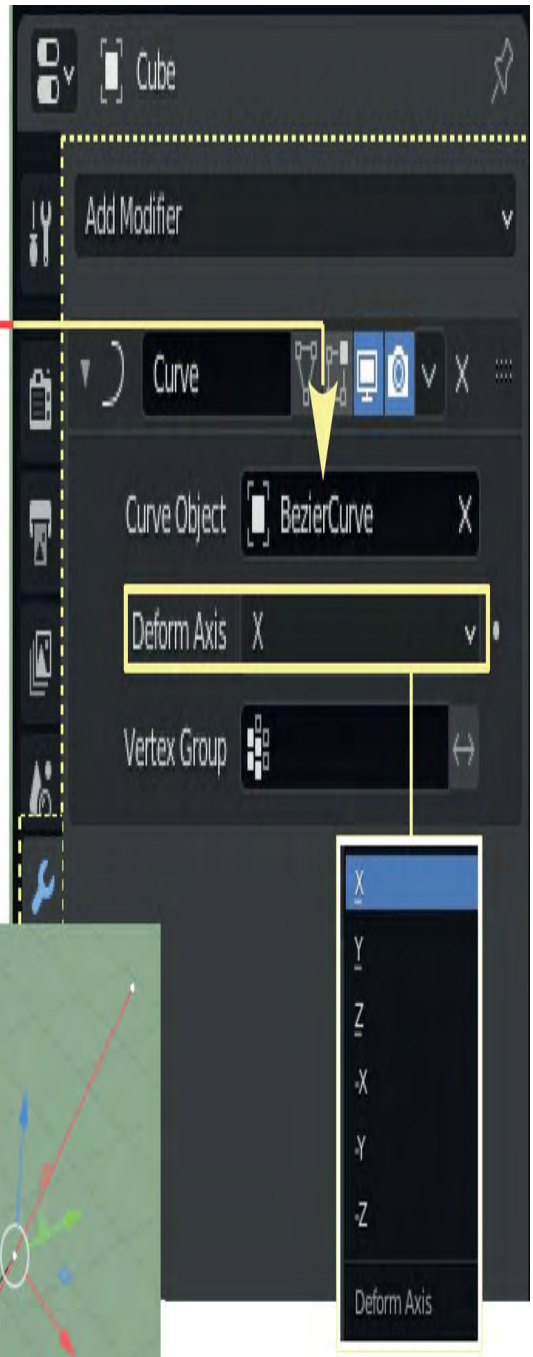


Figure 9.6

Deselect the Curve and **select the Cube.** Be in **Object Mode** and add a **Curve Modifier.** Enter **BezierCurve** as the name in the **Object** panel (Figure 9.7).

Click and select **Bezier Curve** in the menu.



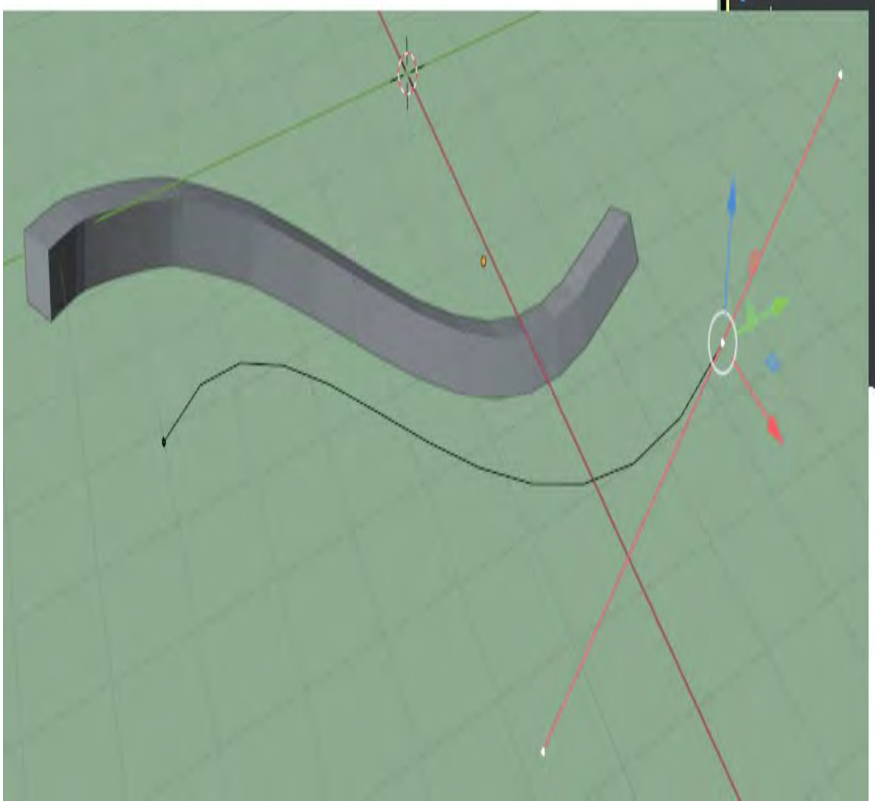


Figure 9.7
Figure 9.8

Manipulate the Curve in **Edit Mode.** Change its shape and position to affect the shape of the Cube (Figure 9.8).

Note: in the Curve Modifier panel the **Deformation Axis** is the **X Axis.**

9.6 Displace Modifier

The **Displace Modifier** manipulates the Vertices of a Mesh Object. If Vertices are assigned to a **Vertex Group** (see Chapter 5 — 5.9) and the group is entered in the Modifier, only the Vertices belonging to the group will be affected. Incorporating a **Texture** displaces Vertices according to the dark and light values in the Texture (see also, Texture Displacement Chapter 17—17.9). Start with the default Blender Scene and replace the **Cube** with a **Plane.** Have the Plane selected in **Edit Mode** and subdivide sixteen times. Select a group of Vertices and create a **Vertex Group** in the Properties Editor, Object Data buttons, see Chapter 5 - 5.9 (Figure 9.9, 9.10).

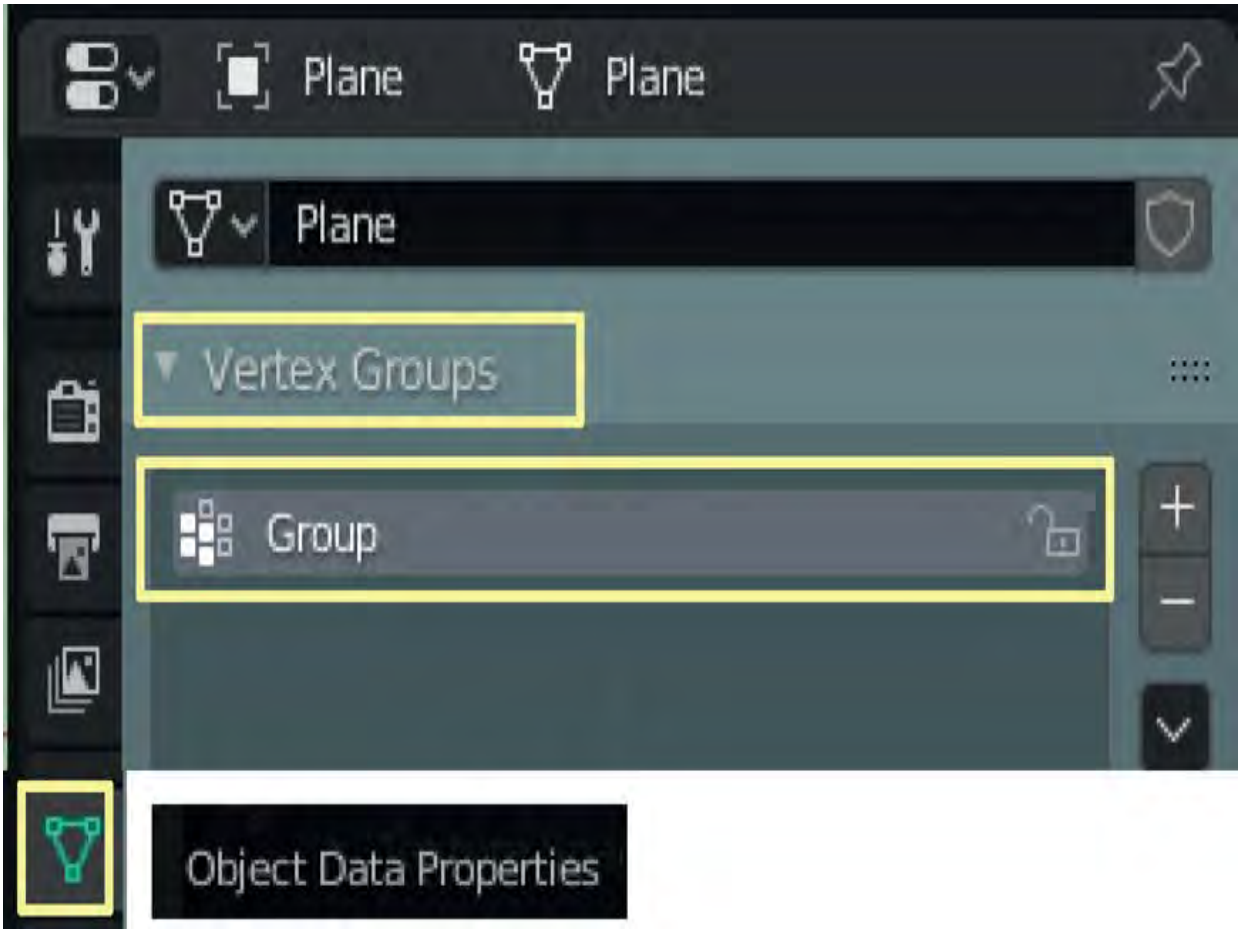

Plane
Plane
Plane
Vertex Groups
Group
Object Data Properties

Figure 9.9

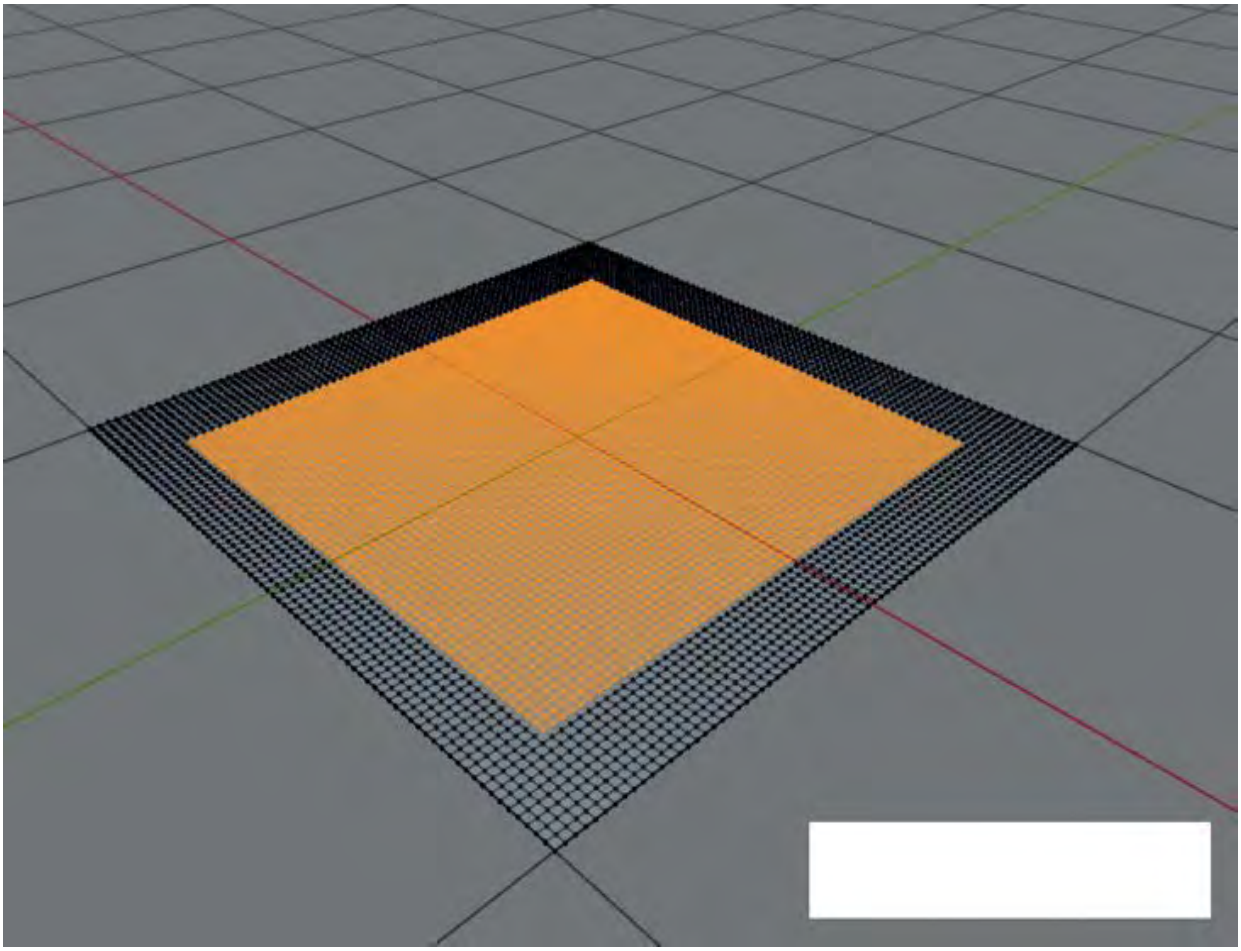


Figure 9.10

Tab back to **Object Mode.** Add a Material (default gray is OK) then in the **Texture Type** add a Noise Texture (Textures are discussed in Chapter 17 but for now, click the Texture button, click **New** and where you see **Type: Image or Movie,** click the bar and select **Noise** in the menu that displays. **Note** the Texture Name in this example is: Texture) (Figure 9.11).

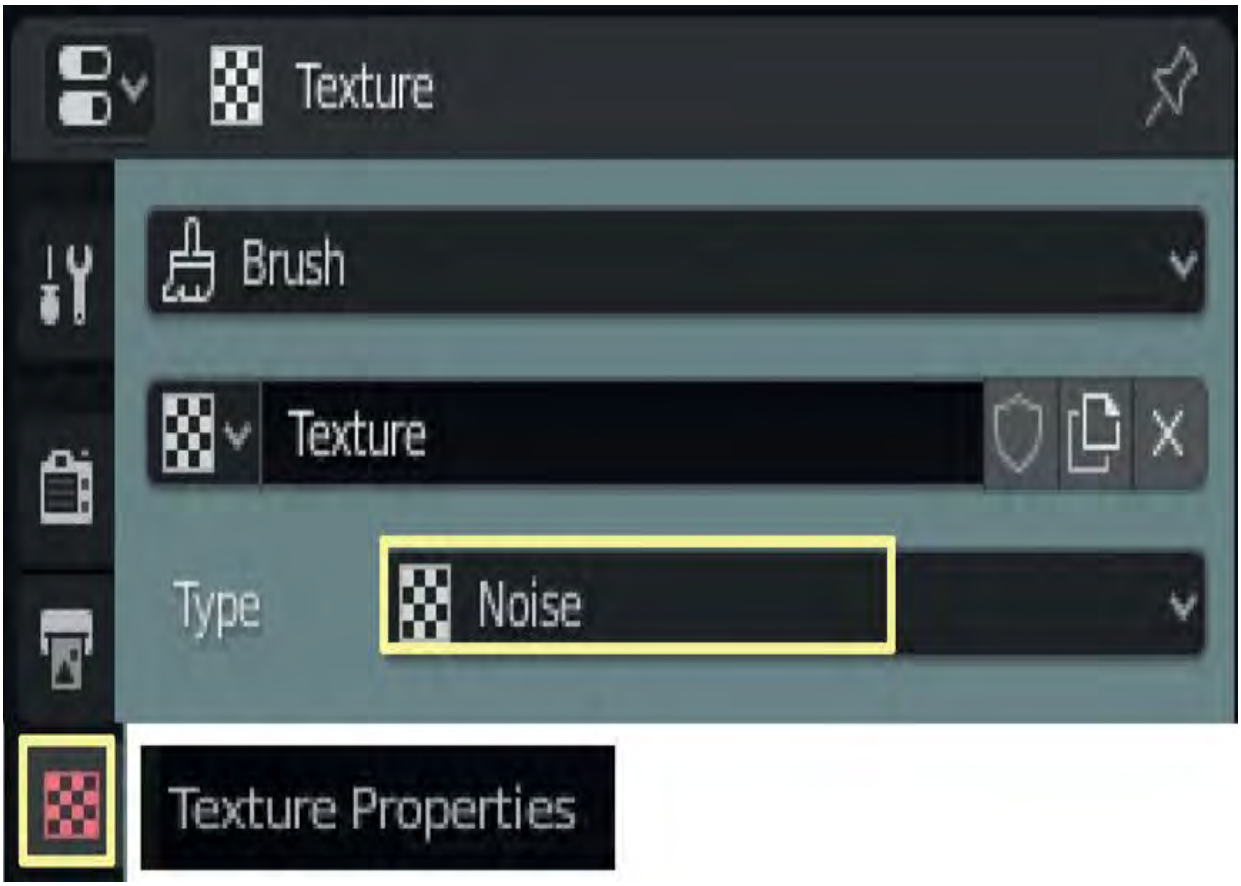


Figure 9.11

In the Properties Editor, with the Plane selected, go to the Modifier buttons and add a **Displace Modifier** (Figure 9.12).

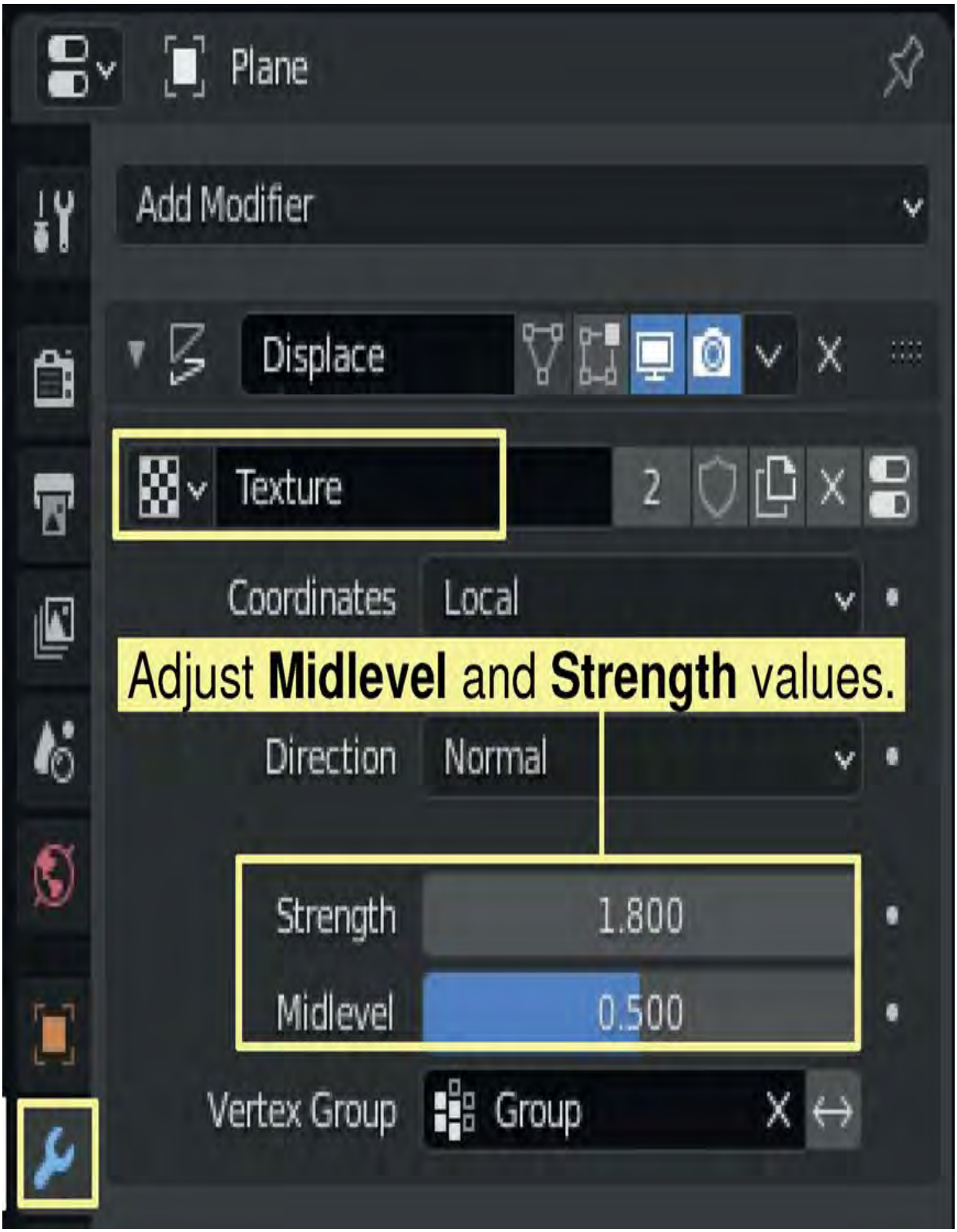

Plane
Add Modifier
Displace
Texture
2
Coordinates
Local
Adjust Midlevel and Strength values.
Direction
Normal
Strength
1.800
Midlevel
0.500
Vertex Group
Group

Figure 9.12

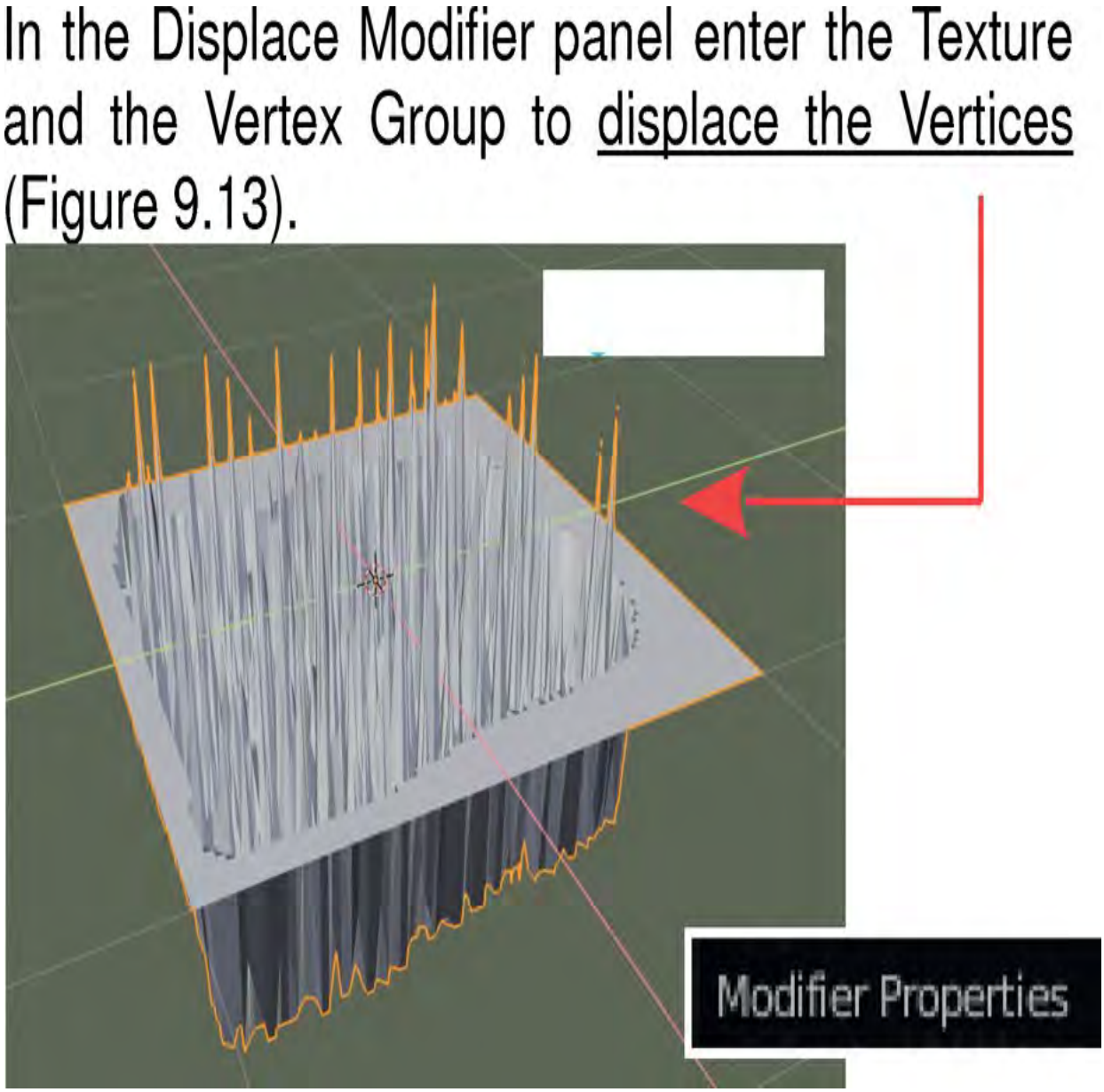


Figure 9.13

9.7 Hook Modifier

The **Hook Modifier** allows you to manipulate or animate selected vertices of a mesh while in Object mode. Vertices are assigned (hooked) to an **Empty** Object which is moved in Object Mode pulling the selected vertices with it. This can be used for a static mesh deformation or the movement can be animated.

Start with the default Scene with the Cube Object selected. Tab to **Edit mode** and select one Vertex (corner) only.

Press **Ctrl + H key** and select **Hook to New Object.** An **Empty Object** is added to the Scene. In the **Properties Editor, Modifiers Properties,** you will see that a **Hook Modifier** has been added named **Hook-Empty** (Figure 9.14).

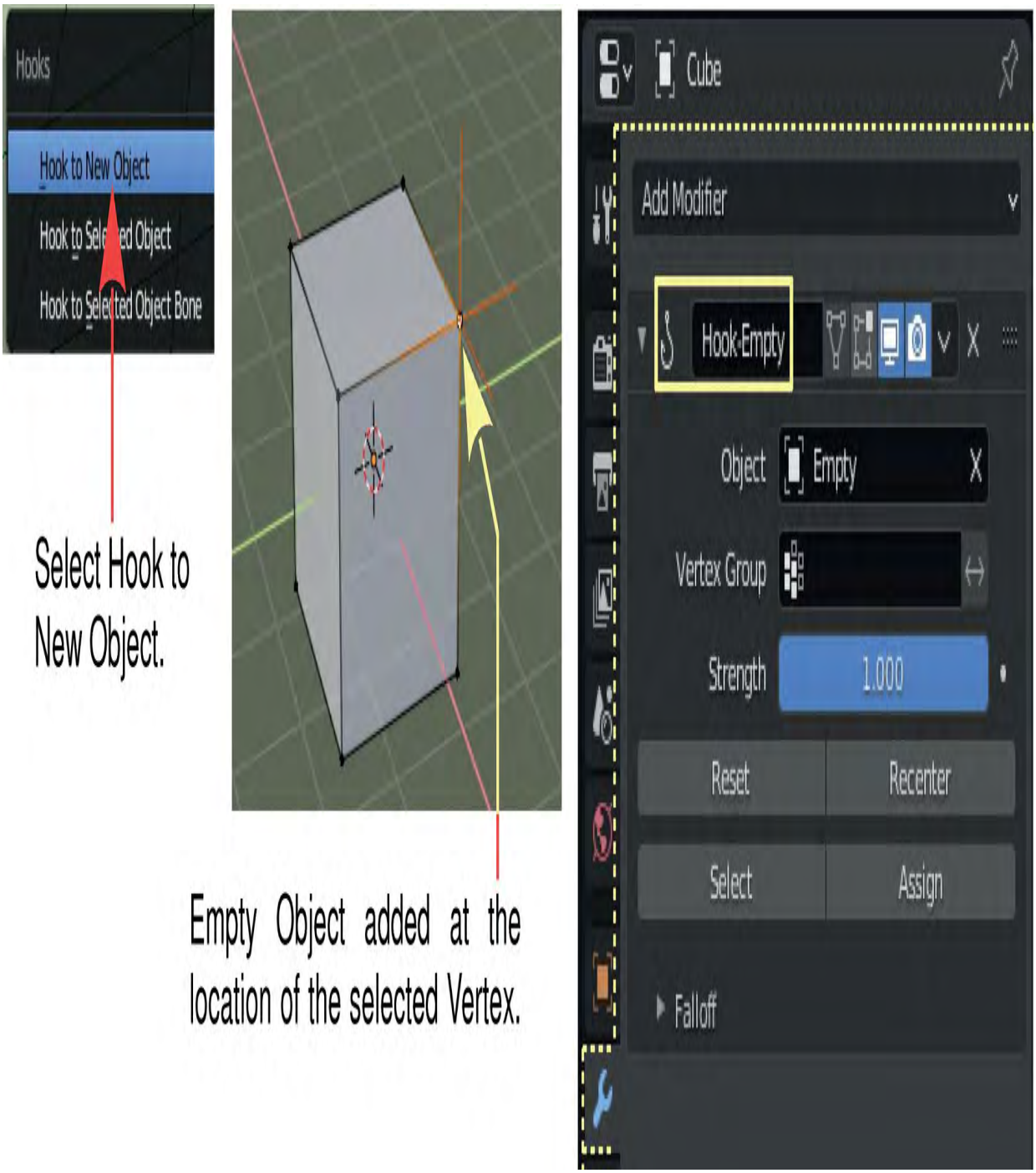


Figure 9.14

Select and move the **Empty** in **Object Mode** to move the Vertex (Figure 9.15, 9.16).

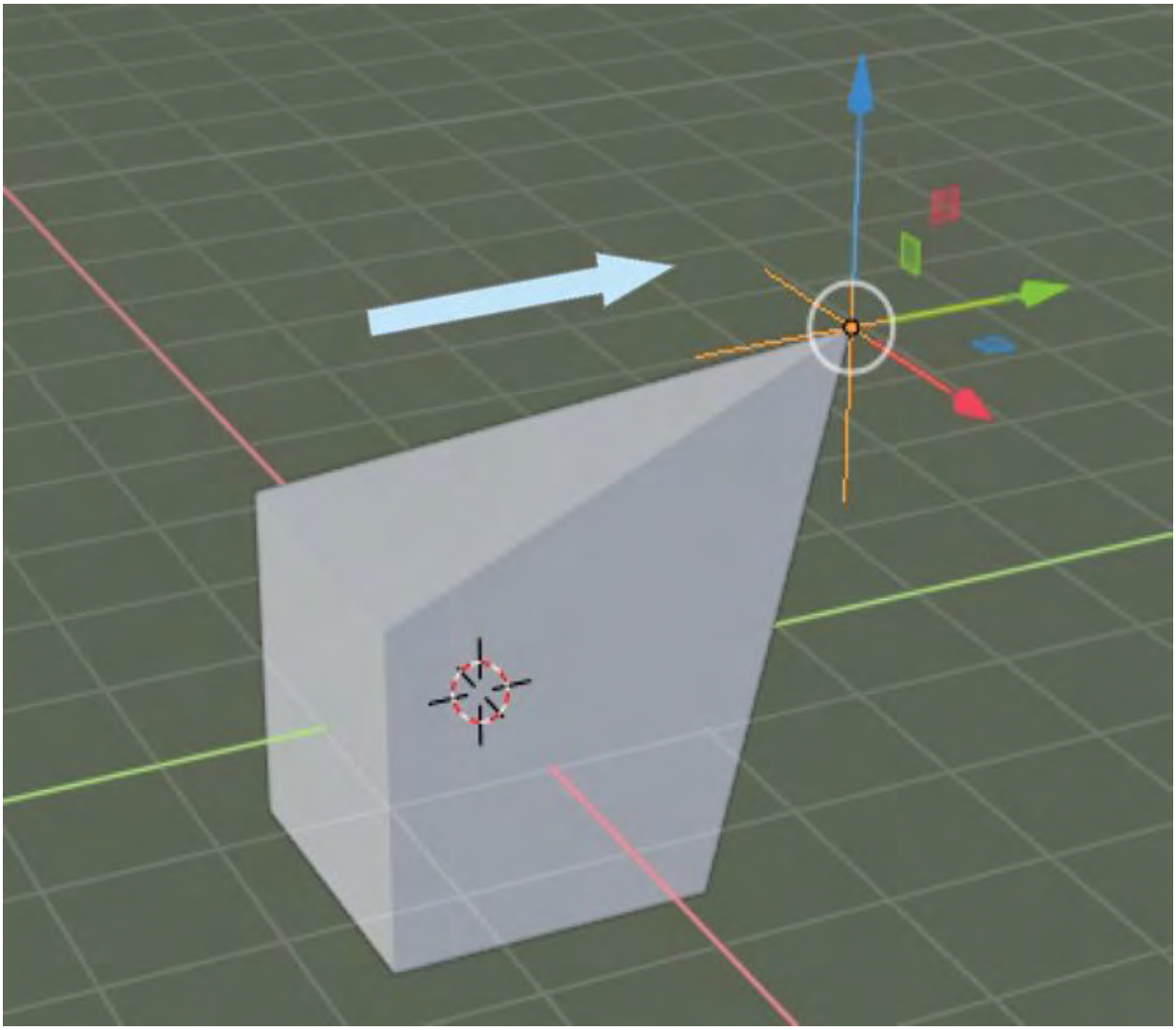


Figure 9.15

In **Edit Mode** you will see the Empty in the new location but the selected Vertex remains in its original position.

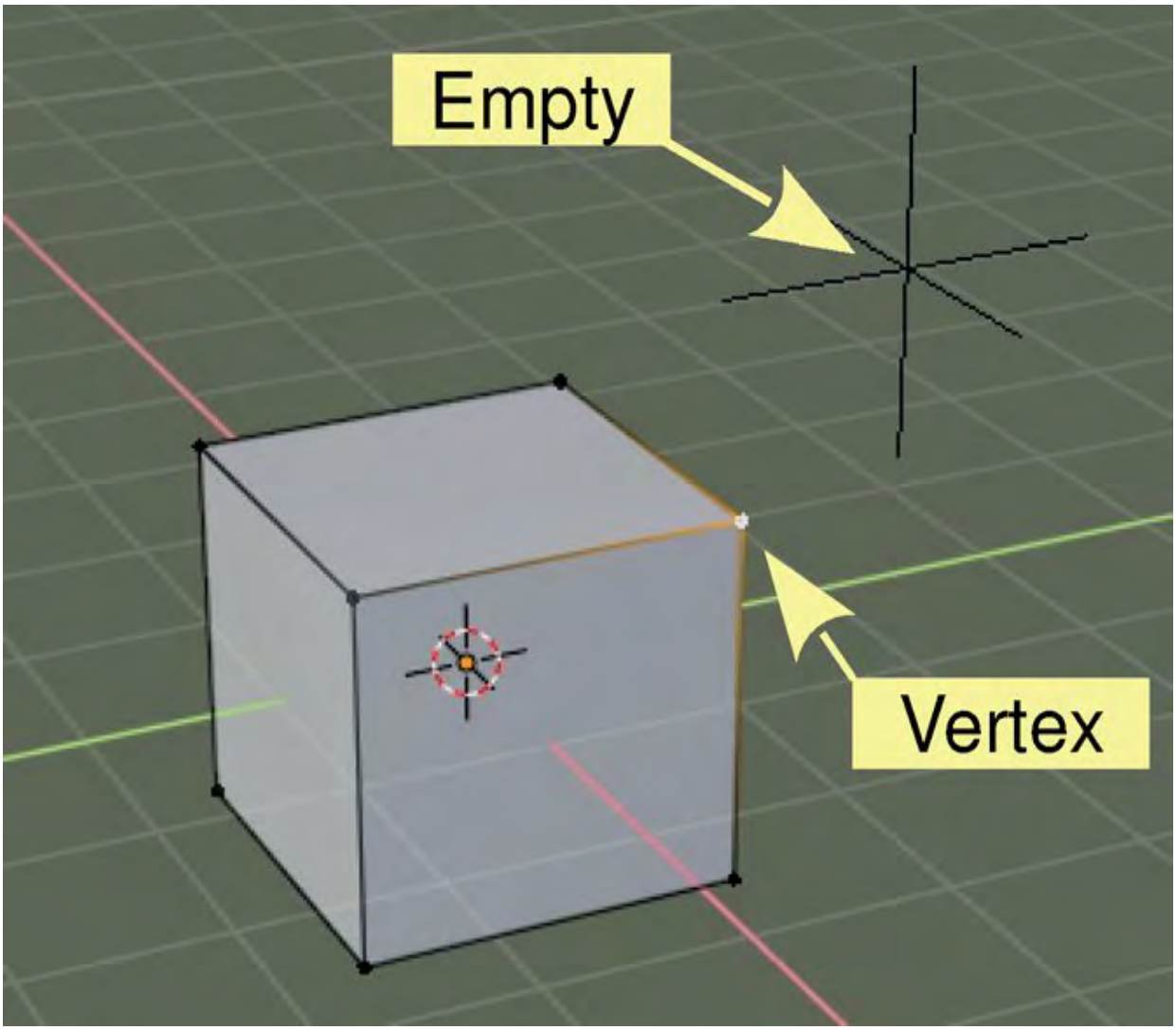


Figure 9.16

In Object mode click **Apply** in the Modifier panel and the Cube deformation is made permanent. You may delete the Empty Object.

Alternative Add Hook Method

With the default **Cube Object** in the default Scene, **Tab** to **Edit Mode** and select one Vertex only. In the **Properties Editor, Object Data Properties, Vertex Groups Tab** click on the plus sign to add a Vertex Group and click **Assign.** This creates a Vertex Group consisting of the one Vertex.

In **Object Mode** deselect the Cube and add an **Empty Object**. Position the **Empty** away from the Cube (Figure 9.17).

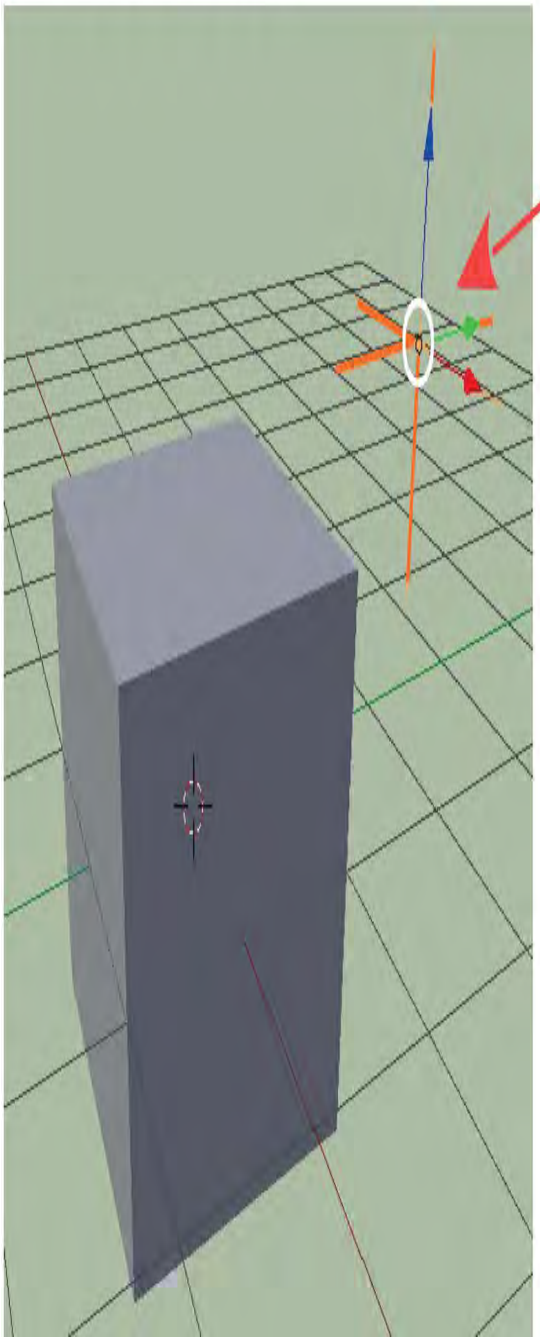


Figure 9.17

Deselect the Empty and select the Cube. In the **Properties Editor, Modifiers Properties** add a **Hook Modifier.** In the **Hook Modifier** panel click on the **Vertex Group panel** and select **Group.** This is assigning the Vertex Group consisting of one single Vertex to the Modifier. Click on the **Object panel** and select **Empty** to assign the Empty Object to the Modifier (Figure 9.18).

By selecting the **Empty** Object in the 3D Viewport Editor and moving it, the single Cube's Vertex in the Vertex Group will follow the movement (Figure 9.19).

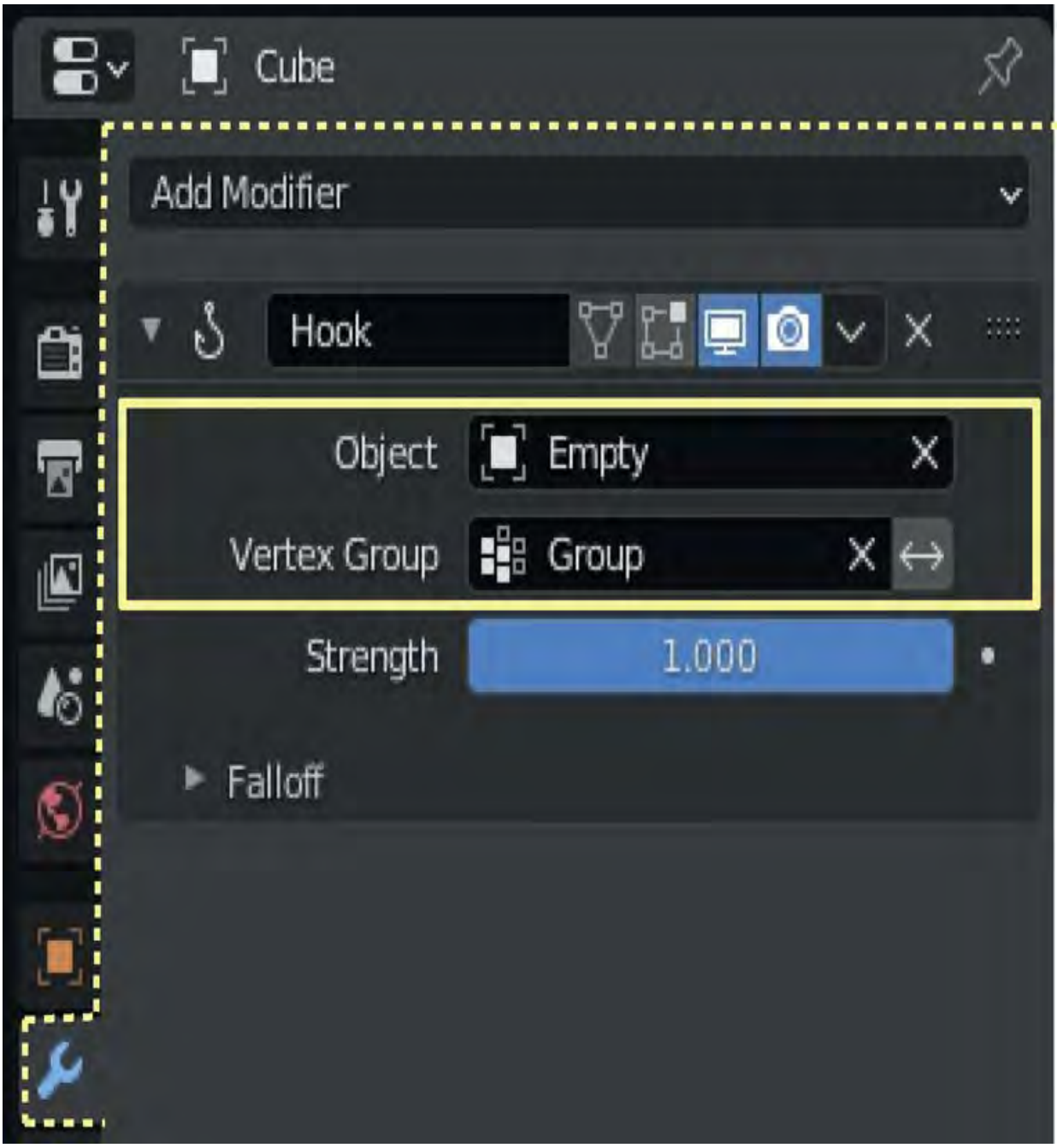

Cube
Add Modifier
Hook
Object
Empty
Vertex Group
Group
Strength
1.000
Falloff

Figure 9.18

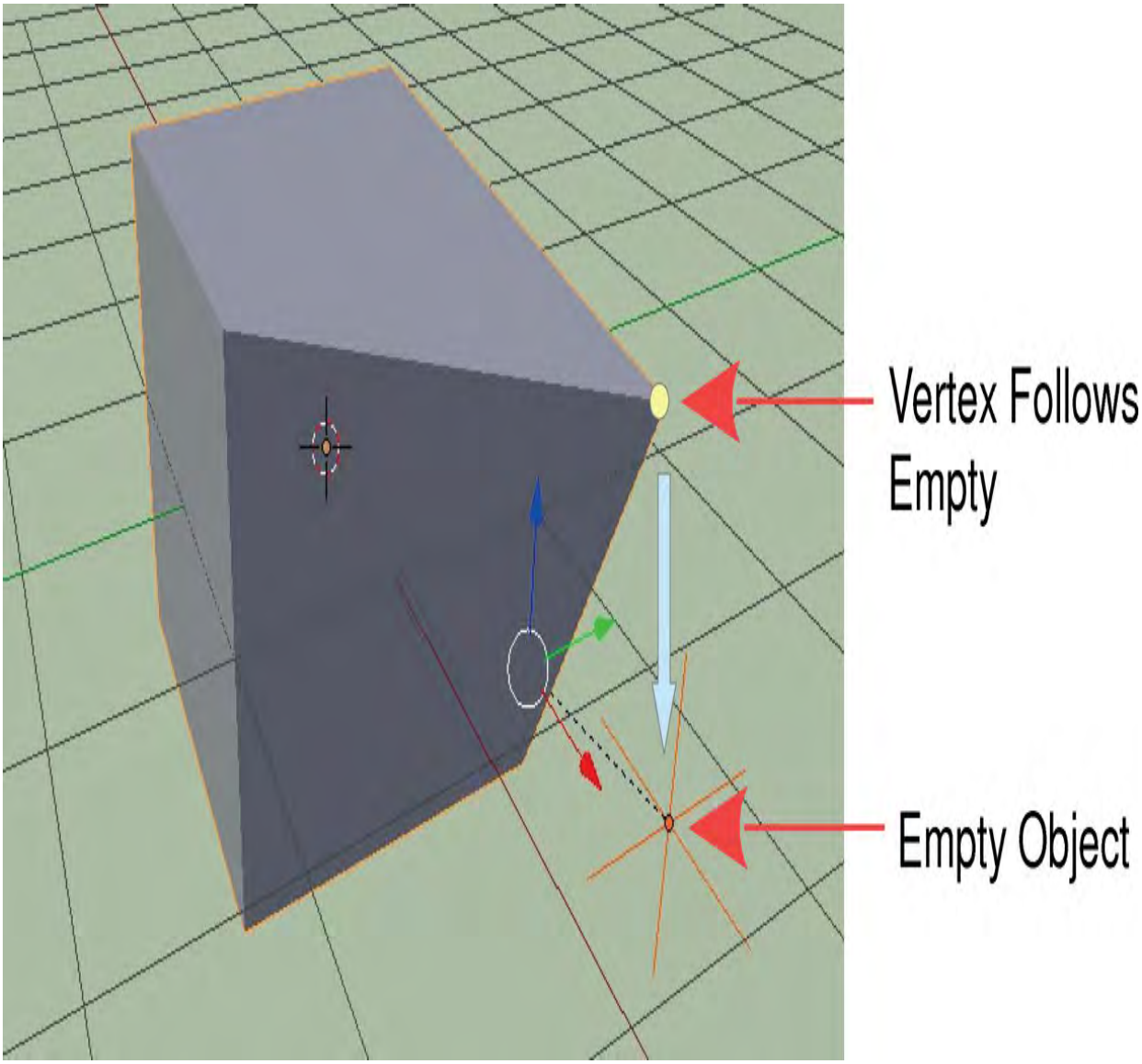


Figure 9.19

9.8 Laplacian Deform Modifie,

The **Laplacian Deform Modifier** allows you to pose a mesh while preserving geometric details of the surface. This means, you may alter the shape of an Object as seen in Object Mode while the original mesh is retained in Edit Mode.

The process is performed by assigning a **Hook-Empty** to a Vertex or group of Vertices and anchoring part of the mesh in place. The Hook-Empty pulls the

mesh out of shape against the anchored Vertices.

To replicate the example shown in Figure 9.20 perform the following procedure paying attention to the sequence of operations.

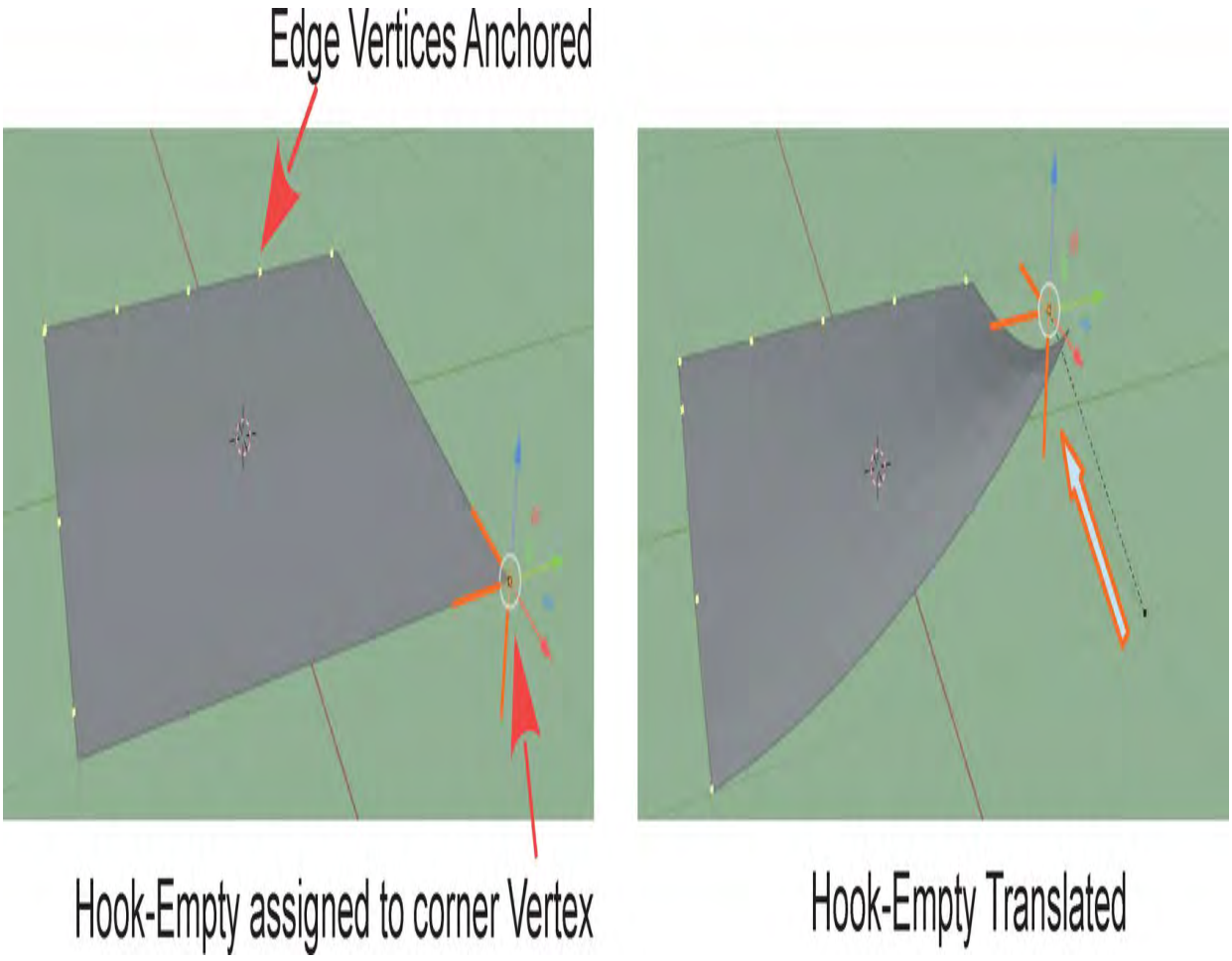


Figure 9.20

Figure 9.20 shows a **Plane** Object with its mesh subdivided ten times in Edit Mode.

While in Edit Mode select the single corner Vertex, press **Ctrl + H Key** and select **Hook to New Object** in the menu that displays. This assigns a **Hook** in the form of an **Empty Object** to the Vertex. In Object Mode you will

see that a Hook-Empty Modifier has been automatically added in the Properties Editor, Modifier Properties.

Have the Plane in the 3D Viewport Editor in **Edit Mode** and with the single Vertex and Hook-Empty remaining selected, select the Vertices forming the two Edges of the Plane opposite the single Vertex with the Hook-Empty (use Circle select Chapter 5 -5.3).

With the Vertices selected, in Edit Mode, go to the **Propertices Editor, Object Data Properties** (Figure 9.21) and create a **Vertex Group** and Assign the Vertices to the Group (Chapter 5 -5.9).

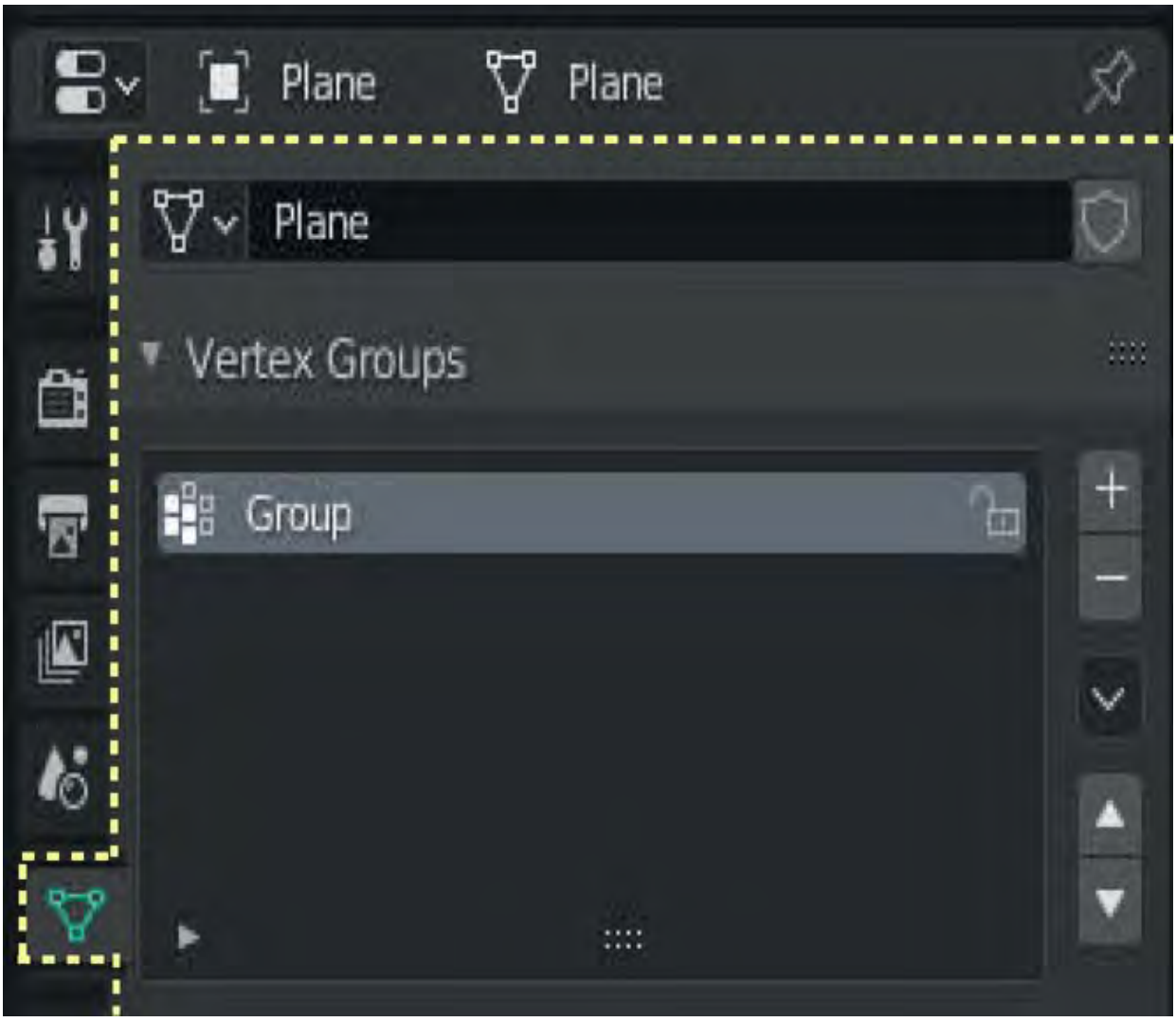
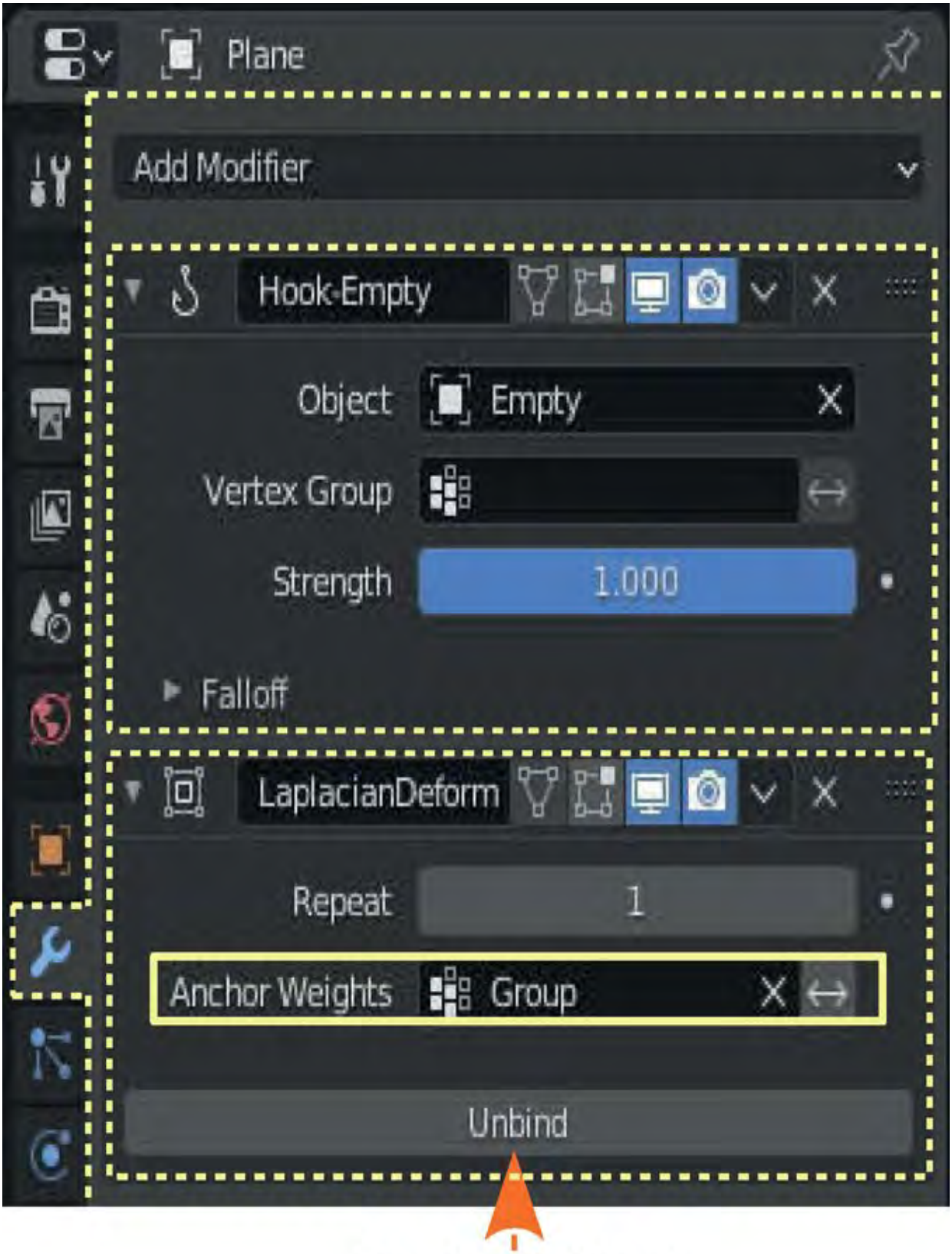


Figure 9.21

Note: The single Vertex with the Hook-Empty attached and the Edge Vertices are all selected when assigning Vertices to the Vertex Group. By default the Vertex Group is named **Group.**

Tab to Object Mode and select the Plane.

In the Properties Editor, Modifier Properties, Add a **Laplacian Deform Modifier.** Enter **Group** (the Vertex Group previously created) in the **Anchor Weights** panel and click Bind (changes to Unbind).


Plane
Add Modifier
Hook-Empty
Object
Empty
Vertex Group
Strength
1.000
Falloff
LaplacianDeform
Repeat
1
Anchor Weights
Group
Unbind

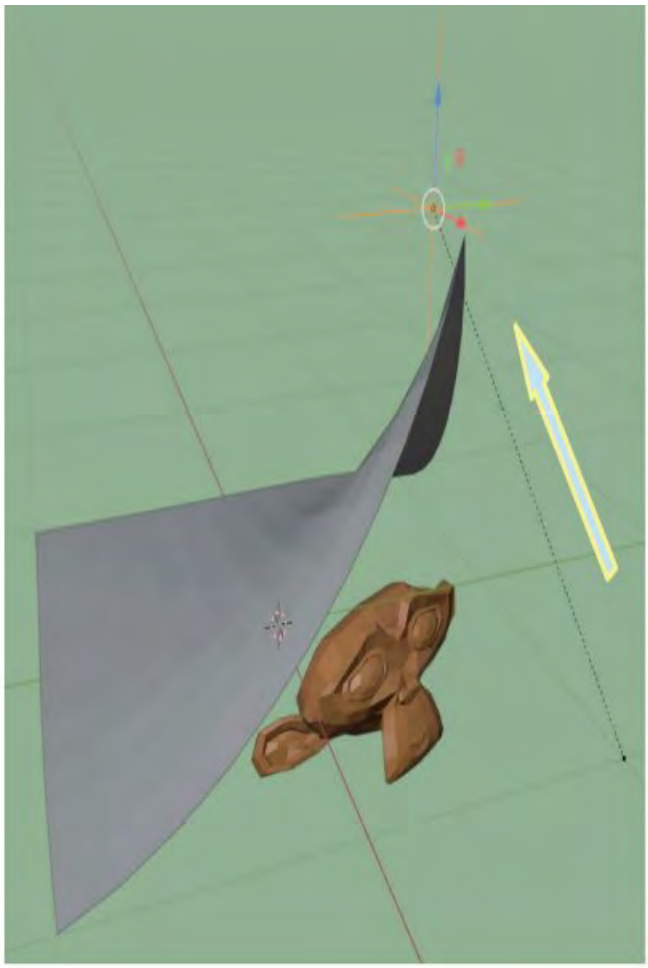
Bind / Unbind

Figure 9.22

By selecting the **Hook-Empty,** pressing the G Key and dragging the Mouse you can move the corner of the Plane with the two opposite Edges remaining in situ.

You could use this to reveal something previously hidden below the Plane.

Animate the Hook-Empty (Chapter 1 - 14)



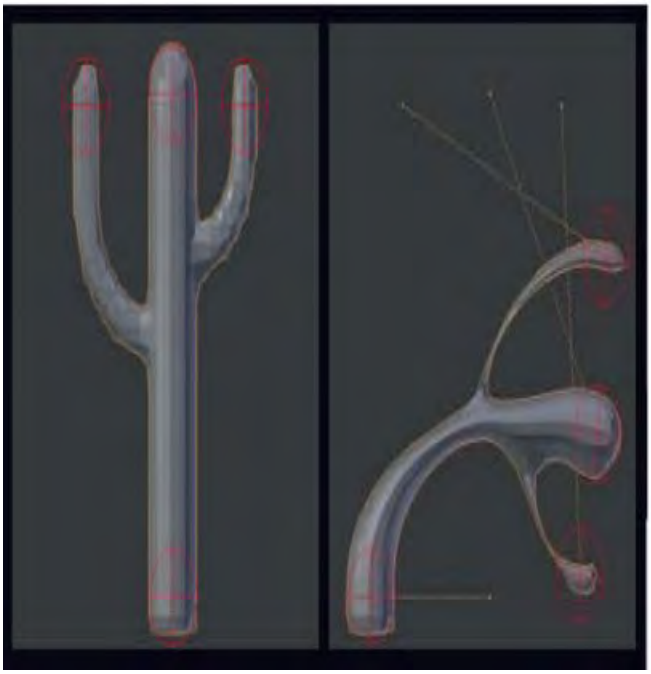


Figure 9.23

9.9 Lattice Modifier

The **Lattice Modifier** is used to deform a mesh Object or to control the **movement of Particles** (Chapter 24). By using the Modifier, it is easy to shape a mesh Object that has many Vertices.

When deforming a Mesh Object there are two components in the Scene; the Object which is to be deformed and a **Lattice.** The Lattice is a special non-renderable Object or, if you like, a non-renderable grid of Vertices. You can use the same Lattice to deform several Mesh Objects by giving each Object a Modifier pointing to the Lattice.

The basic procedure is to surround the Mesh Object or Objects with the Lattice then add a Latice Modifier to each Object pointing to the Lattice. Bear in mind the Lattice does not necessarily have to encapsulate an Object.

To demonstrate a **UV Sphere** Object will be deformed using a **Lattice Modifier.**

The Object which is to be deformed must have a reasonable number of Vertices for the Modifier to be effective.

In the default Scene, delete the Cube Object and add a UV Sphere. Remember, when Objects are added to a Scene they are located at the position of the 3D Cursor. By default the 3D Cursor is located at the center of the Scene.

Move the UV Sphere to one side. Moving the sphere is purely for demonstration purposes.

In the Properties Editor you will find a **Lattice Object Properties** button.

Lattice Object Data Properties

Figure 9.24

Figure 9.25

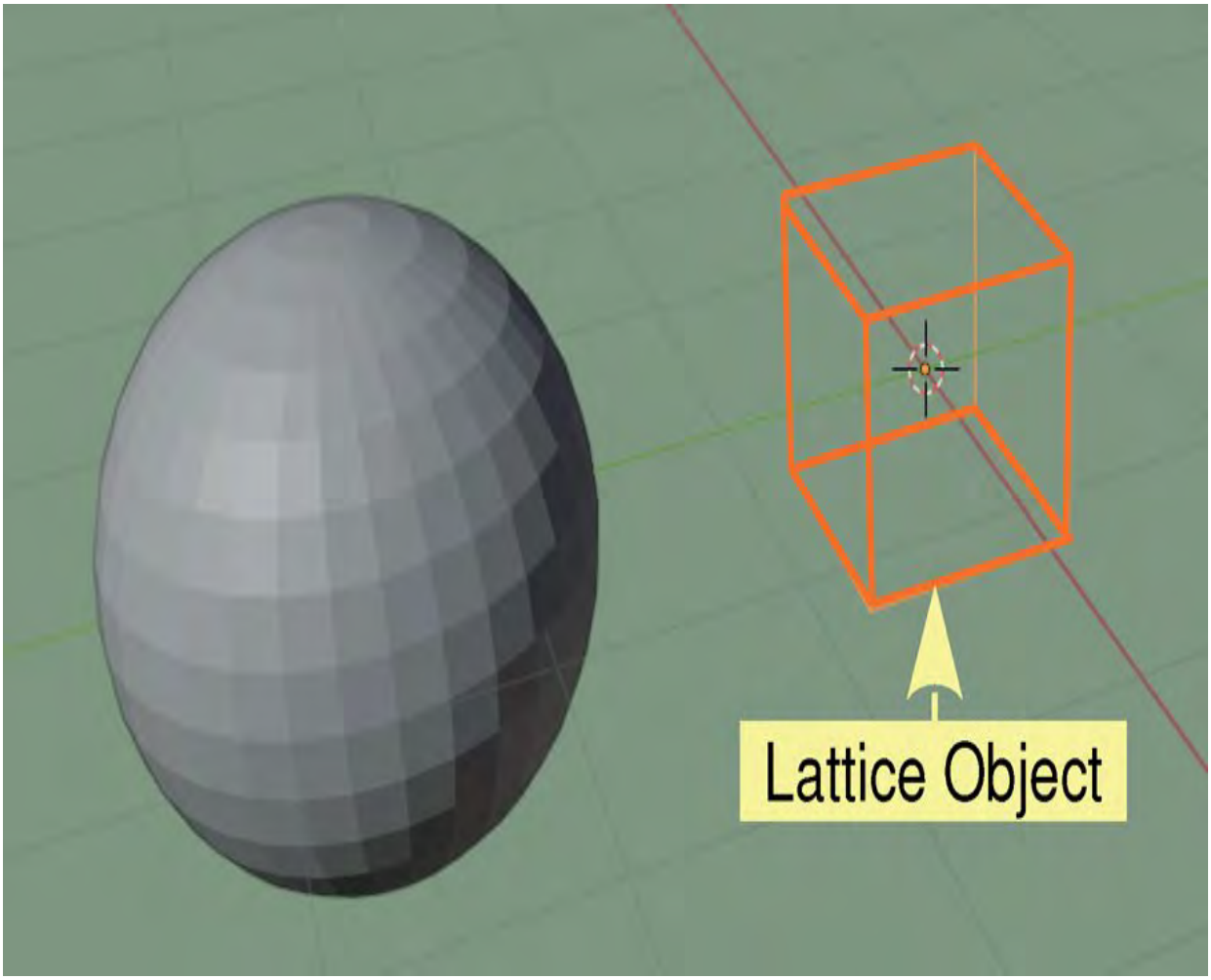


Figure 9.26

Note that the UV Sphere and the Lattice are entered at default Scale: 1.000 although the Lattice displays at half the size of the Sphere.

In this example a Lattice will encapsulate the Object being deformed, therefore, scale the Lattice up and position it around the UV Sphere (Figure 9.27).

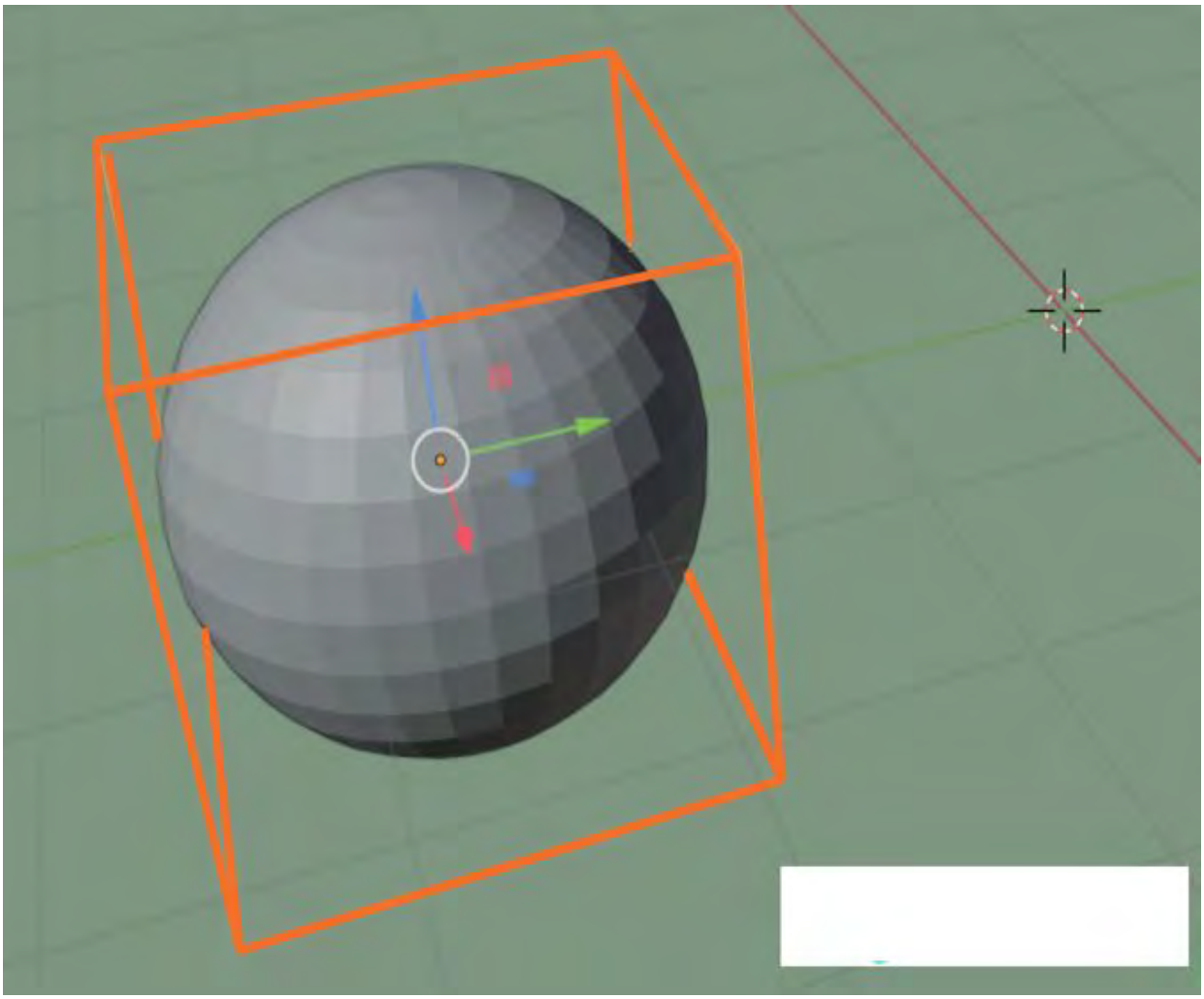


Figure 9.27

Tab into Edit Mode. You may subdivide the Lattice by increasing the **Resolution U, V and W** values in the **Lattice Object Data Properties** panel (Figure 9.23 Resolution values all: 2). How many subdivisions depends on the detailed control required in the deformation. Change the Resolution to: U, V and W = 3.

In Object Mode, select the UV Sphere and Add a **Lattice Modifier** in the Properties Editor (Figure 9.28). Enter Lattice in the Object panel of the Modifier to point to the Lattice.

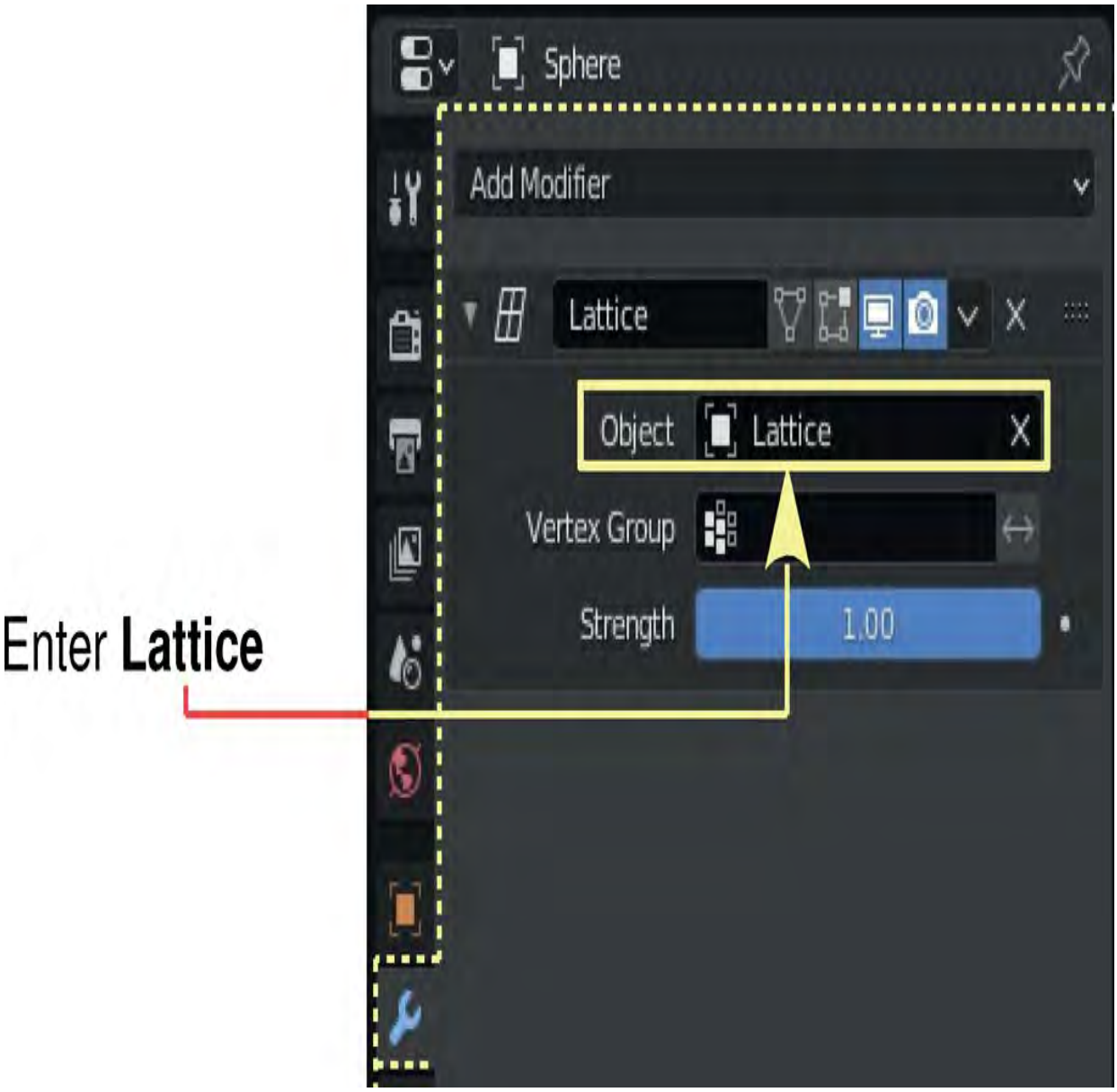
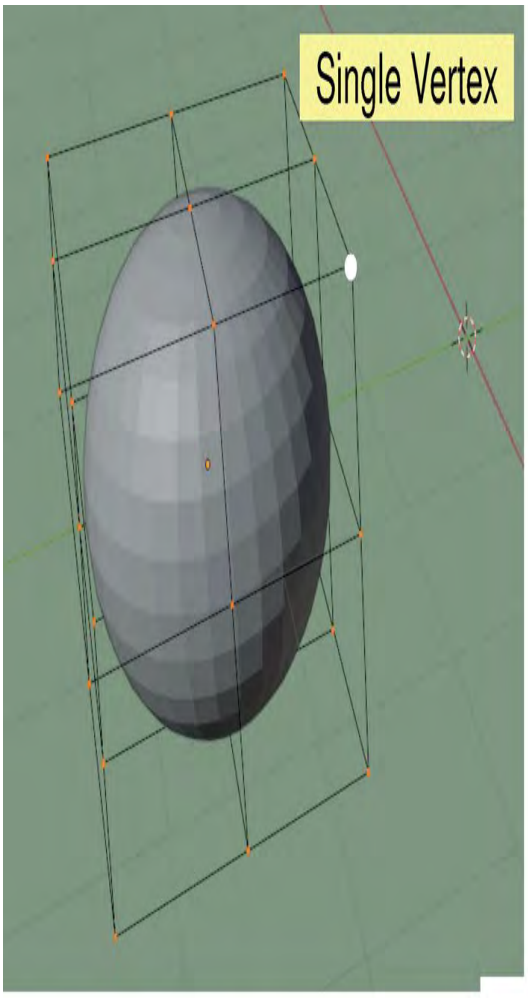
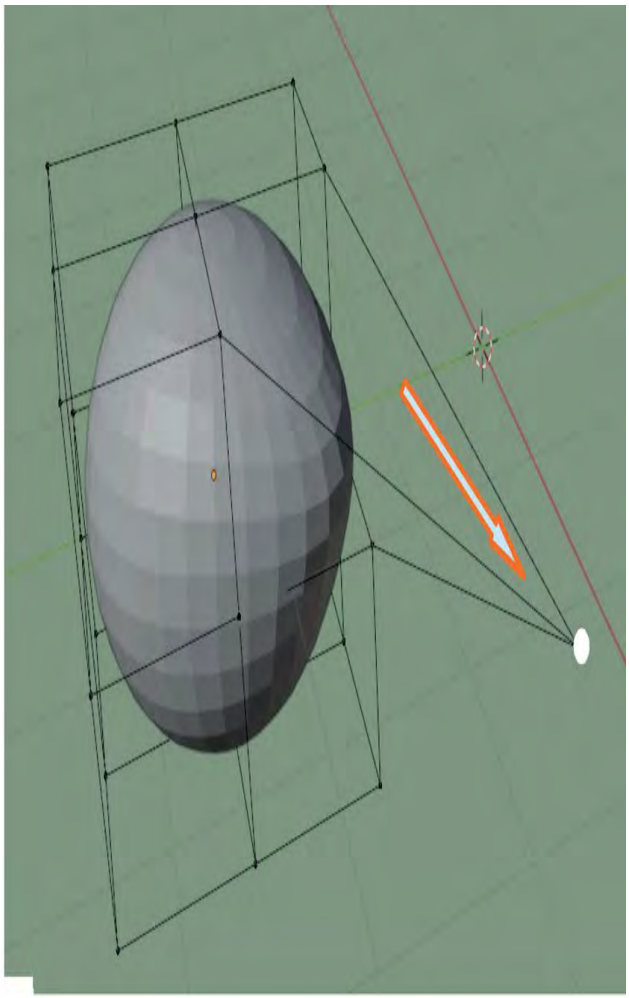
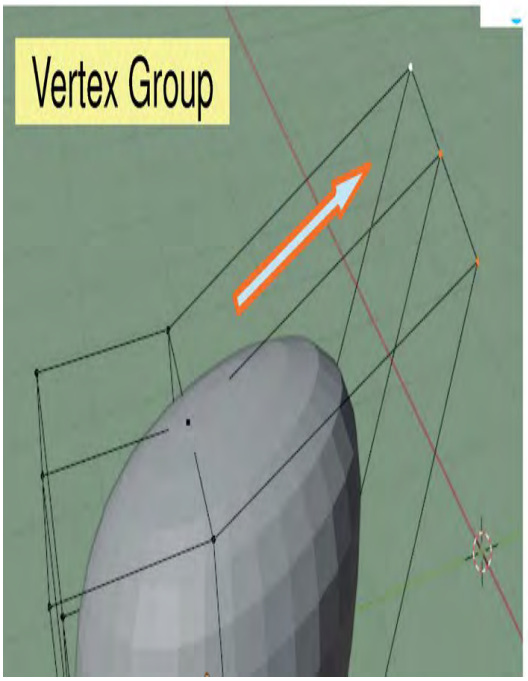


Figure 9.28

Deselect the UV Sphere, select the Lattice and Tab to Edit Mode (Figure 9.29). Select a single Vertex. Press G Key (Grab) and Translate the Vertex to see the UV Sphere deform (Figure 9.29).


Single Vertex




Vertex Group

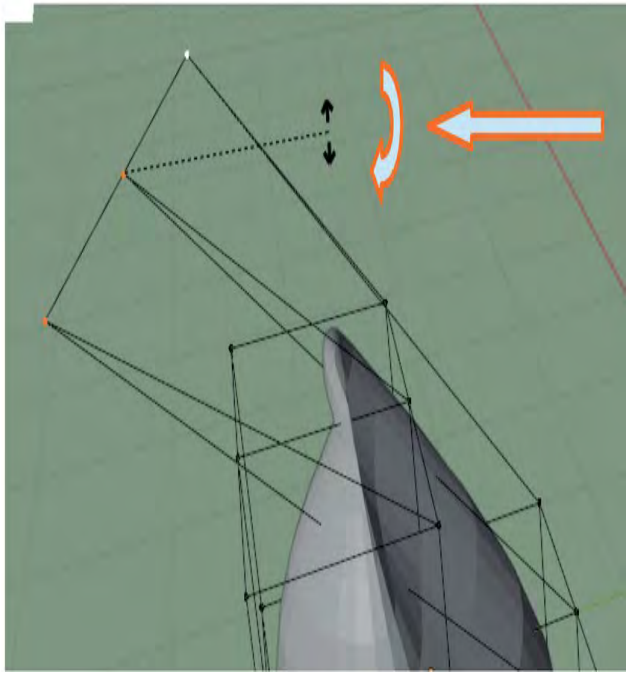


Figure 9.29

You may select multiple Lattice Vertices to achieve the deformation required. You will have to experiment with different configurations to become proficient in the use of the Lattice Modifier. Figure 9.30 (over page) shows a single Lattice at the center of the Scene with two Objects either side to which Lattice Modifiers have been Added. Selecting a single Vertex on the LH Side of the Lattice affects the LH Object and vice versa.

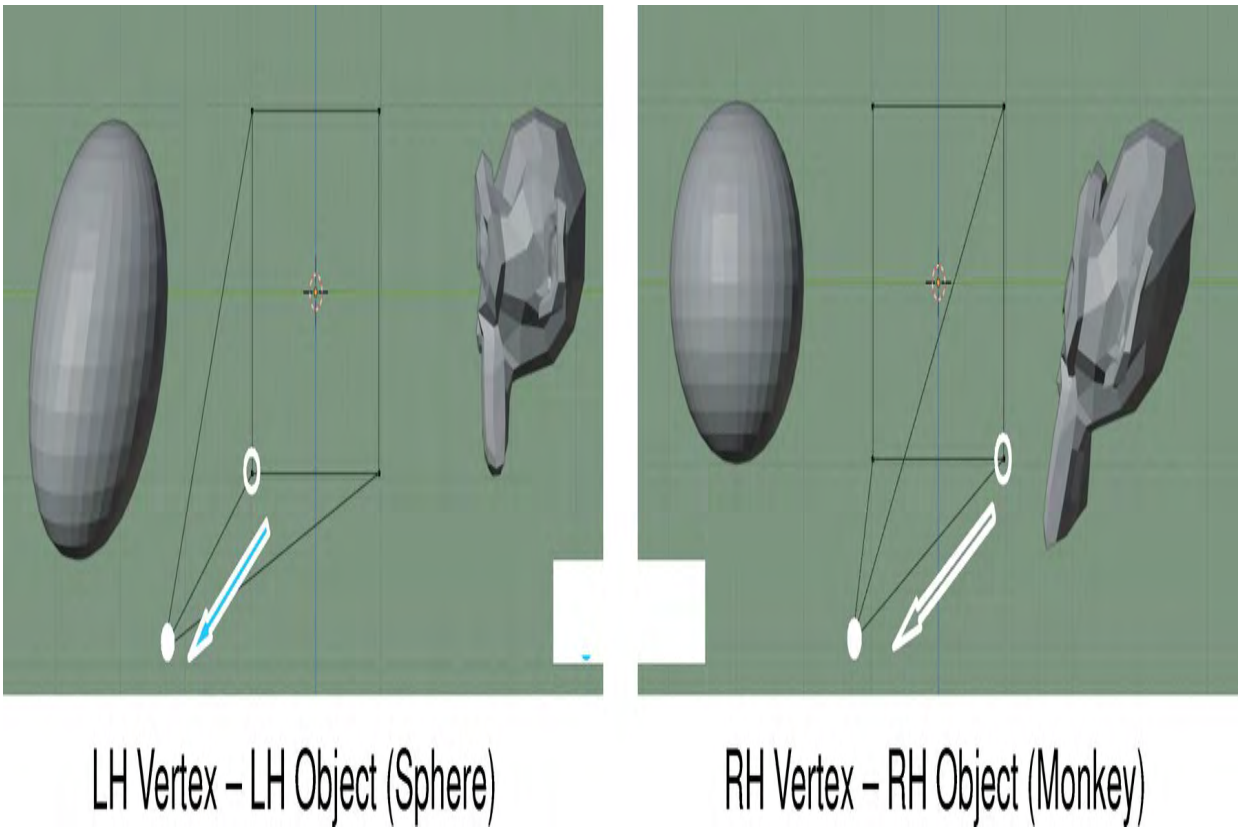


Figure 9.30

9.10 Mesh Deform Modifier

The **Mesh Deform Modifier** deforms a mesh with **Cage Mesh.** This is similar to a Lattice Modifier but instead of being restricted to the regular grid layout of a lattice, the cage can be modeled to fit around the mesh Object

being deformed. The Cage Mesh must form a closed cage around the part of the mesh to be deformed, and only vertices within the cage will be deformed. Typically the cage will have far fewer Vertices than the mesh being deformed.

Model a UV Sphere Object by scaling on the Y Axis as shown in Figure 9.31, surround it with a simple cage mesh by scaling a Cube to fit around the elongated Sphere, then select Vertices in **Edit Mode** and extrude (Figure 9.31).

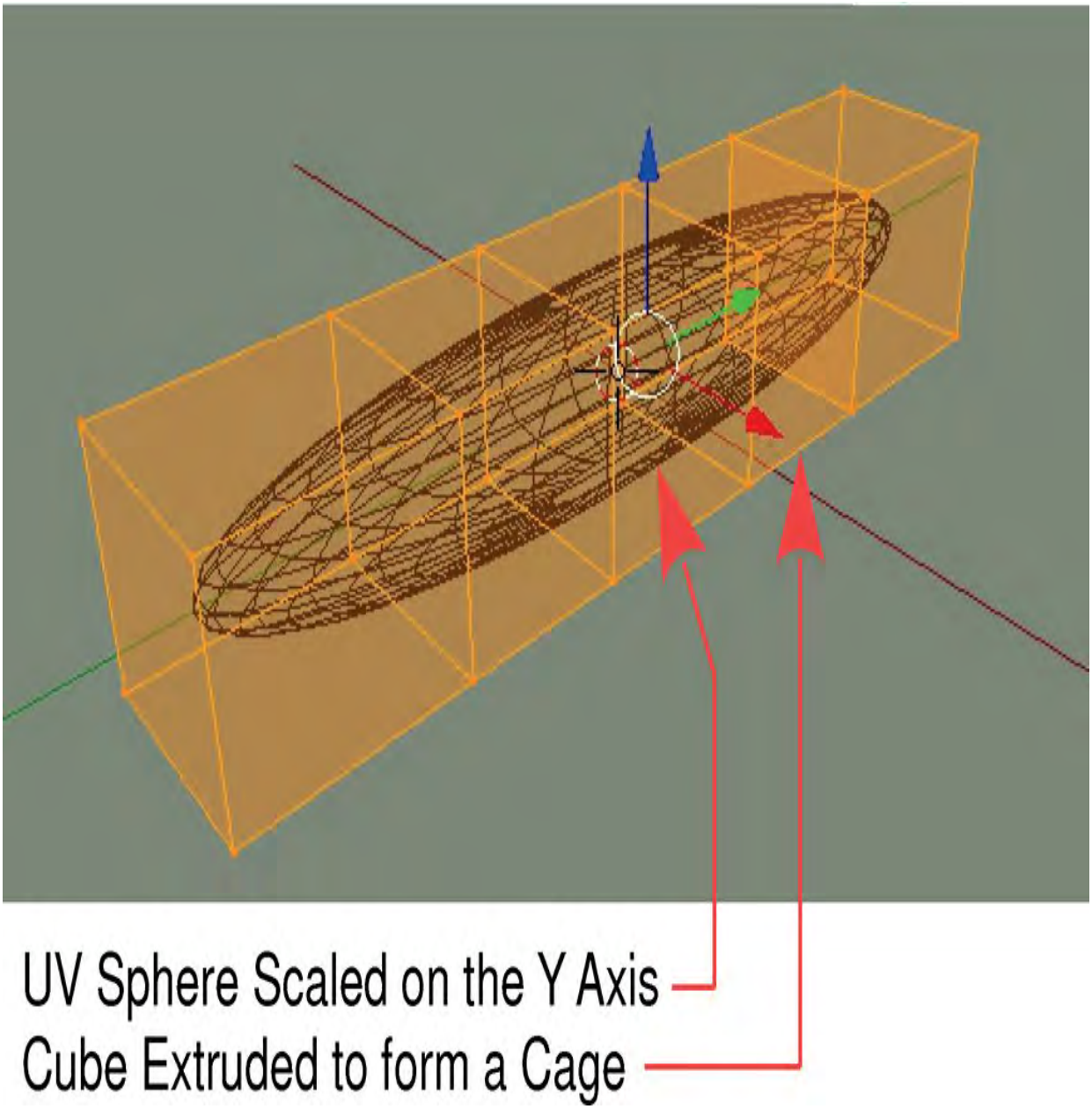


Figure 9.31

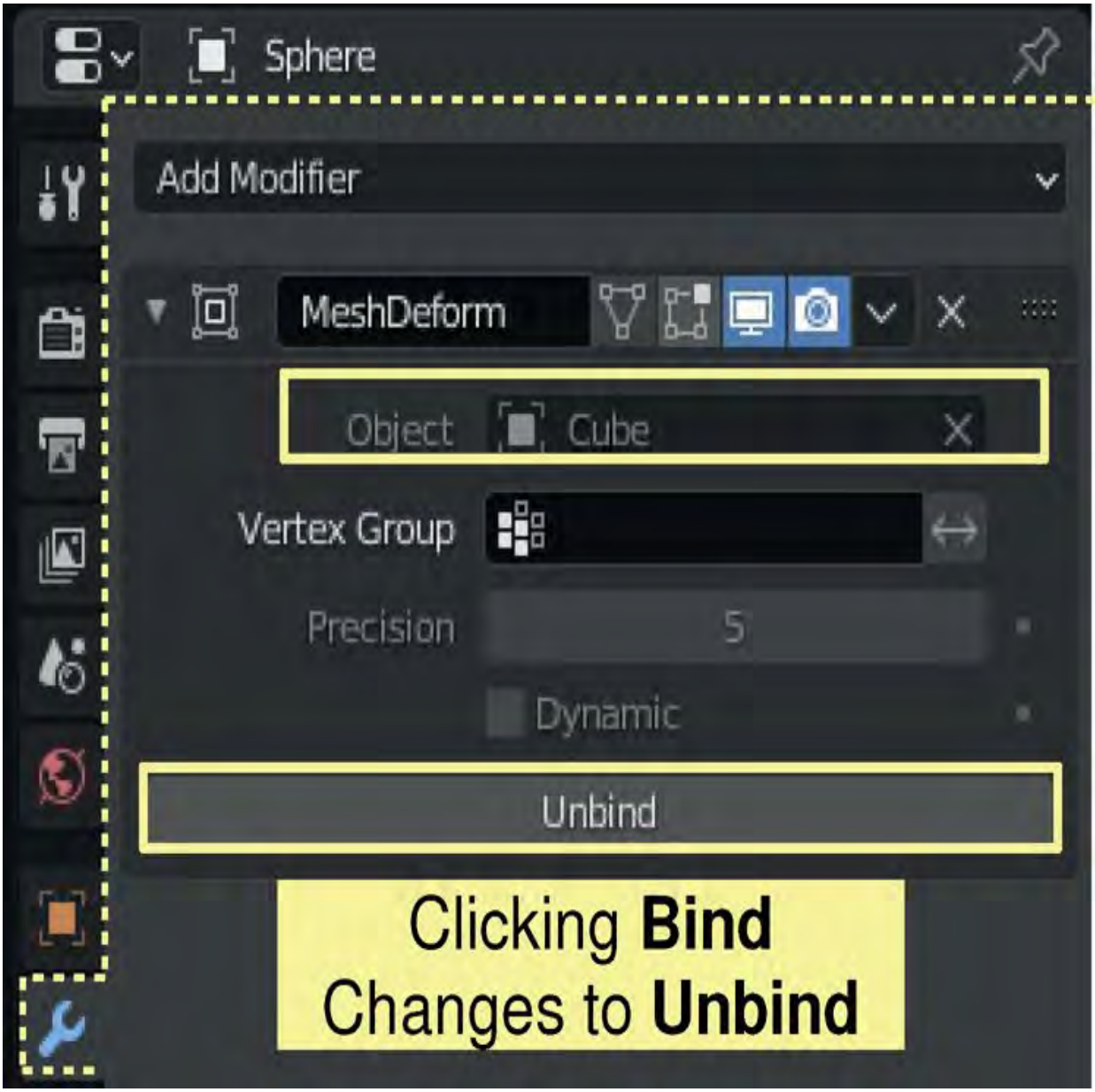

Sphere
Add Modifier
MeshDeform
Object
Cube
Vertex Group
Precision
5
Dynamic
Unbind
Clicking **Bind**
Changes to **Unbind**

Figure 9.32

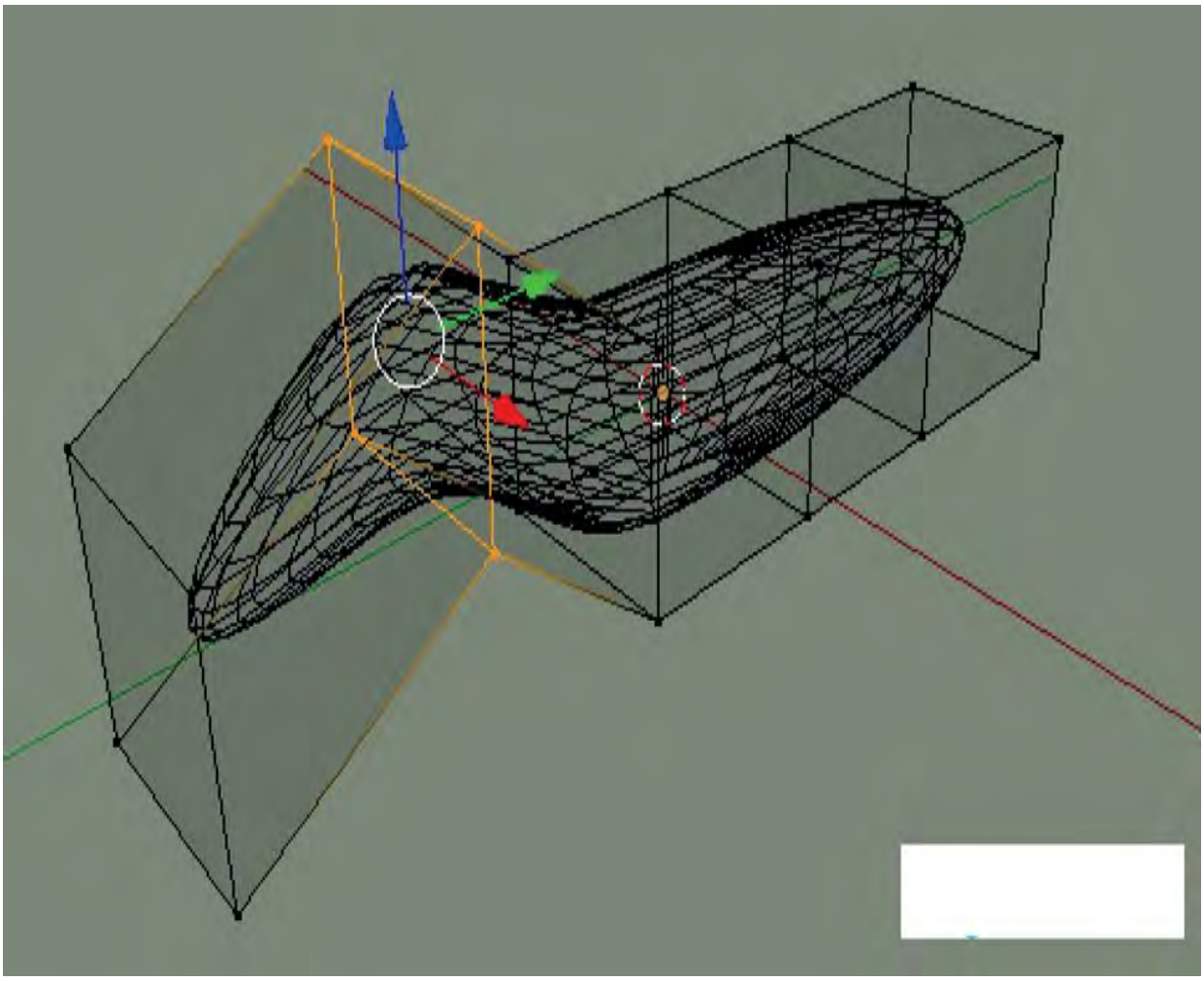


Figure 9.33

Add a **Mesh Deform Modifier** to the scaled **UV Sphere.** Enter the name of the cage mesh (Cube) in the **Object panel** and press **Bind** (Figure 9.32) to link the two meshes. The Bind operation may take several seconds to calculate depending on the complexity of your model. Wait until **Bind** changes to **Unbind** before selecting Vertices on the cage (Figure 9.32). By Moving, Scaling and Rotating the selected Vertices, the

Sphere mesh will be deformed (Figure 9.33). The proximity of the cage to the original Object has an influence on how the deformation reacts.

Note: The Cage Mesh will render in the Scene; With the **Sphere** selected in Object Mode **Apply the Modifier and delete the Cage.**

Click and select Apply in the menu.
Properties
Editor
Modifier
Sphere
Add Modifier
MeshDeform
Object
Cube
Apply
Apply As Shapekey
Save As Shapekey
Duplicate
Move to First
Move to Last

Figure 9.34

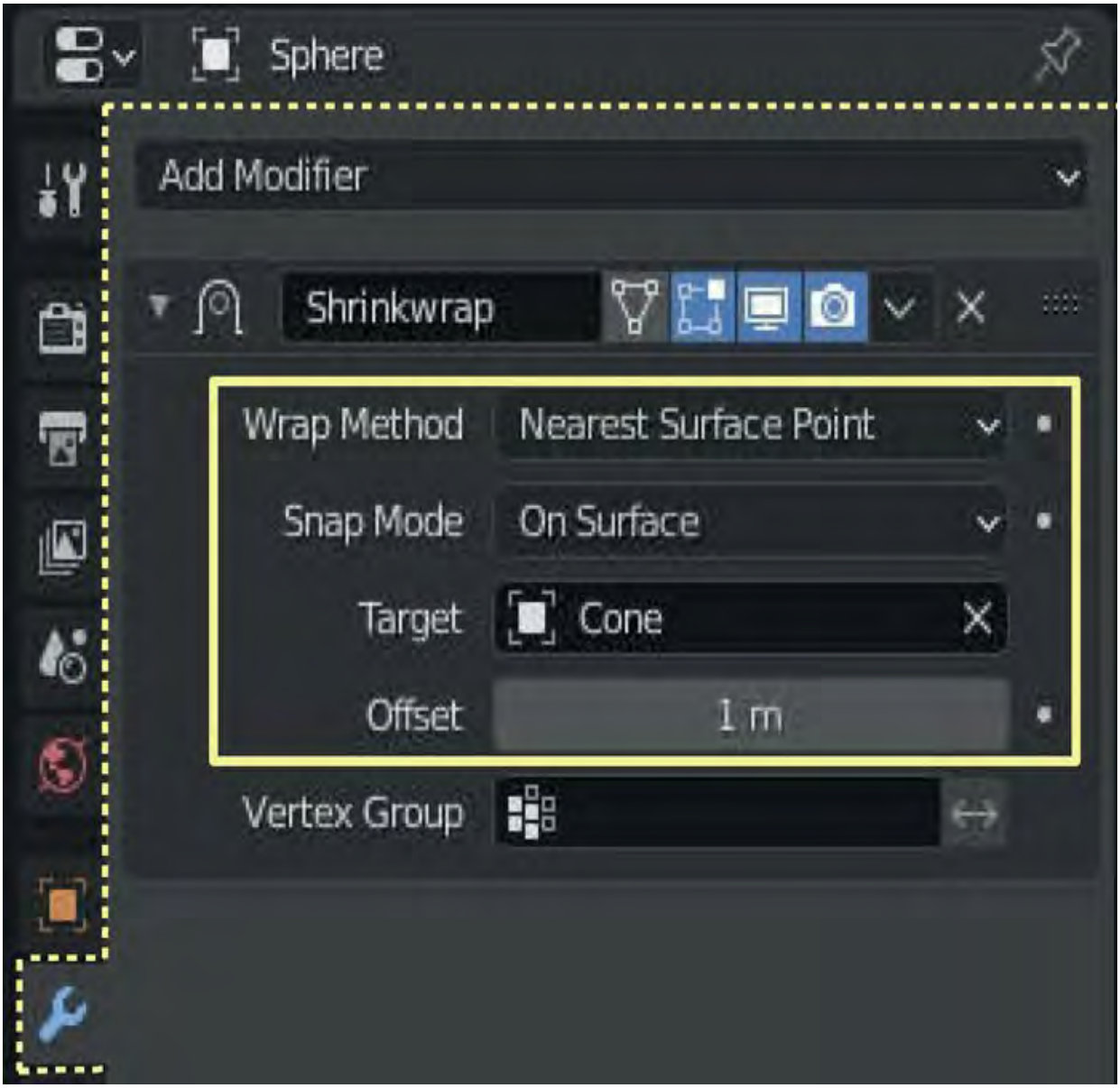


Figure 9.35

9.11 Shrinkwrap Modifier

The **Shrinkwrap Modifier** takes a mesh and shrinks it down, wrapping the mesh around another Object. The deformed mesh can then be **Offset** to produce shapes in between the original shape and the deformed shape.

Delete the Cube in the default Blender Scene and add a **UV Sphere** and a **Cone** mesh Object. The Cone should be located inside the UV Sphere (Figure 9.36), which is easy to see when both Objects are viewed in **Wireframe**

Mode (Chapter 5 — 5.2). Add a **Shrinkwrap Modifier** to the UV sphere, and enter **Cone** in the **Target panel** (Figure 9.35).

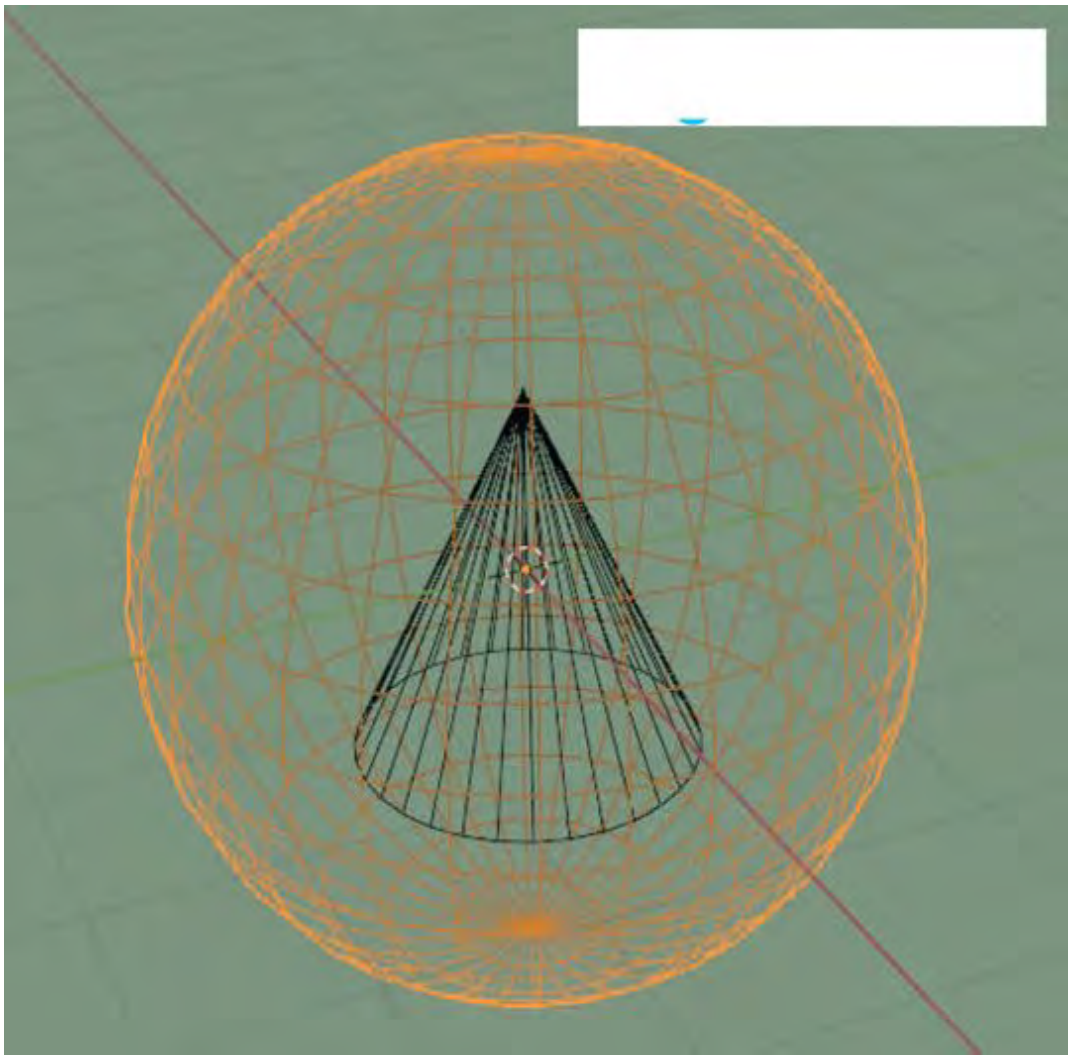


Figure 9.36

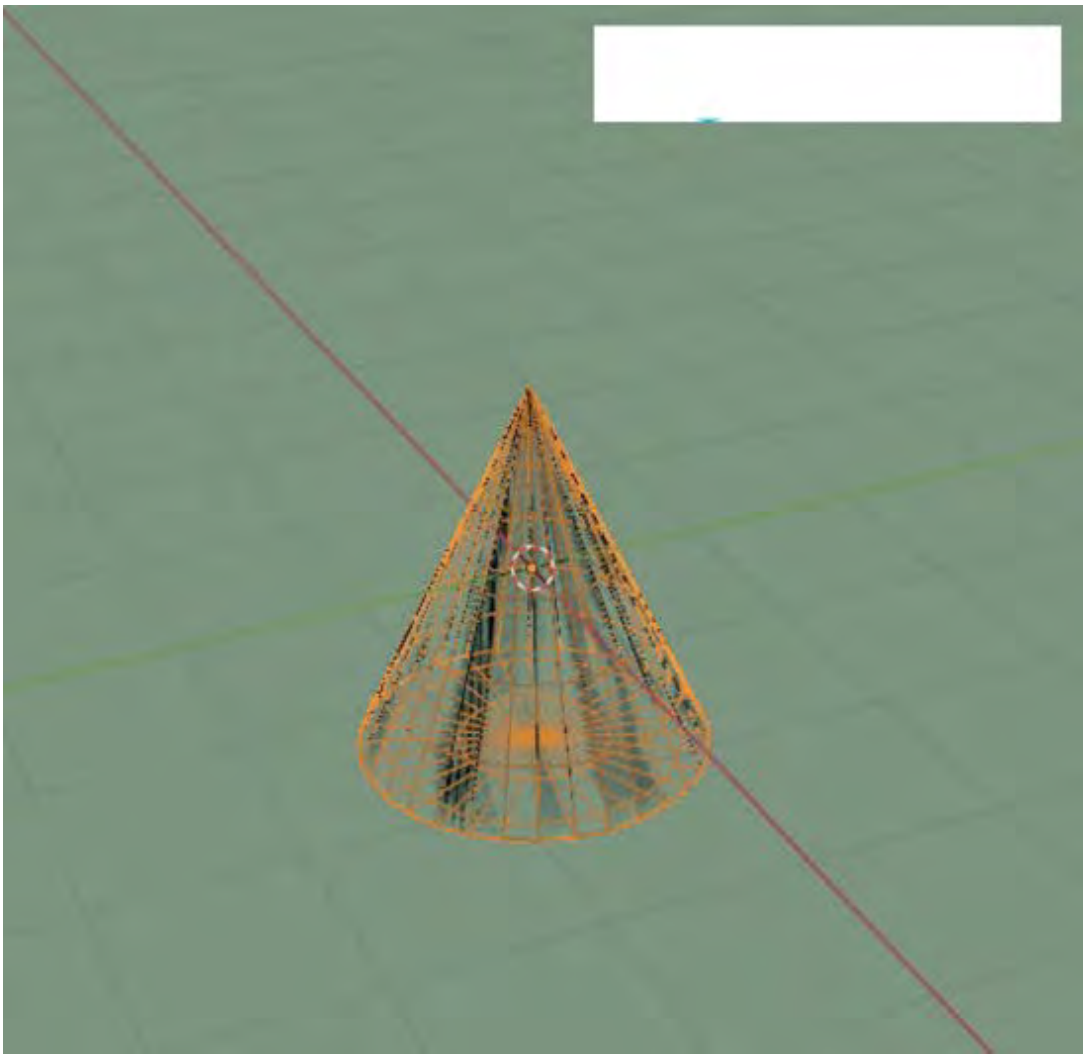
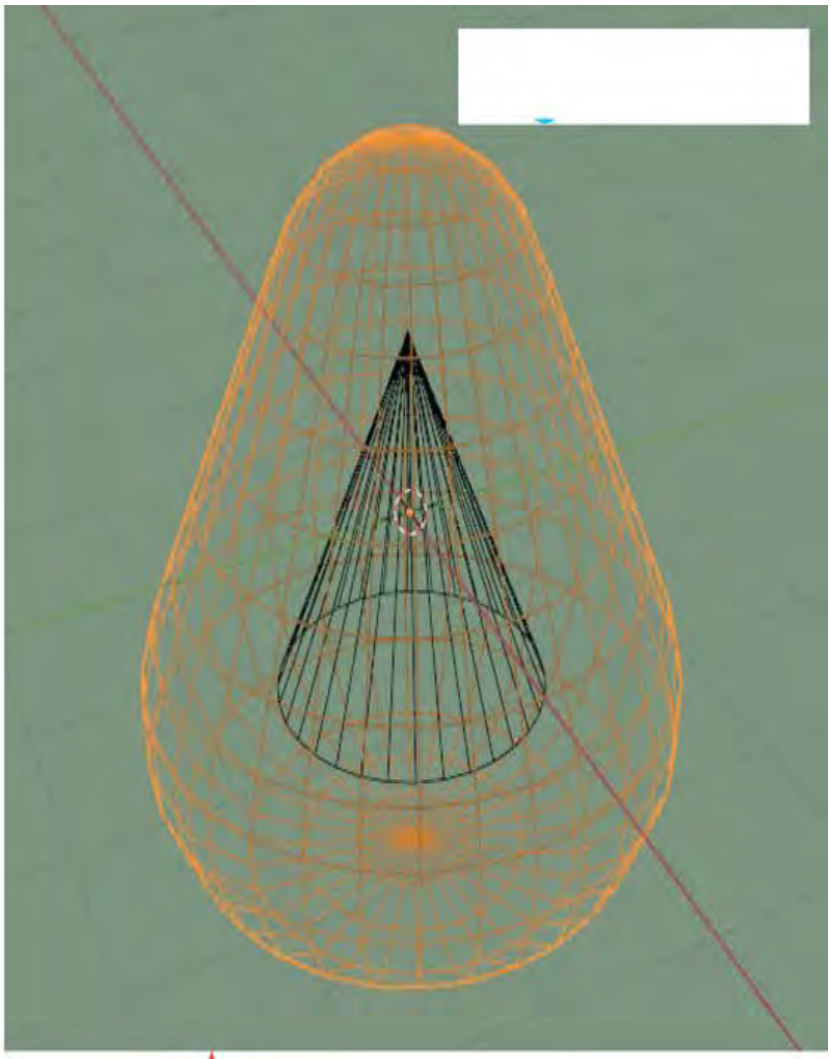


Figure 9.37



Altering the **Offset** value in the Modifier panel changes the size of the modified UV Sphere.

Figure 9.38

Cone inside the UV Sphere Shrinkwrap Modifier added to in VVireframe Display Mode. the UV Sphere. **Target: Cone**

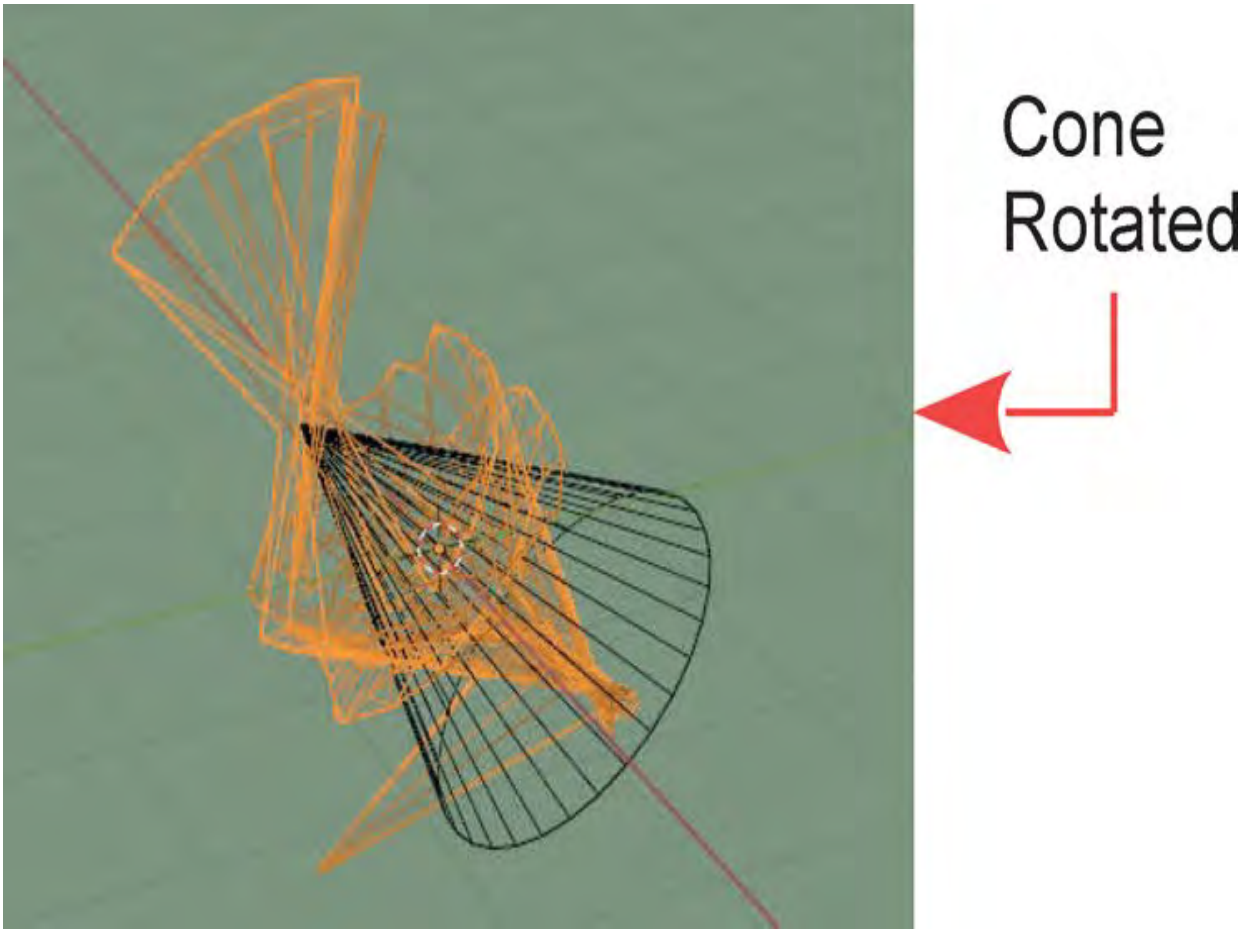


Figure 9.39

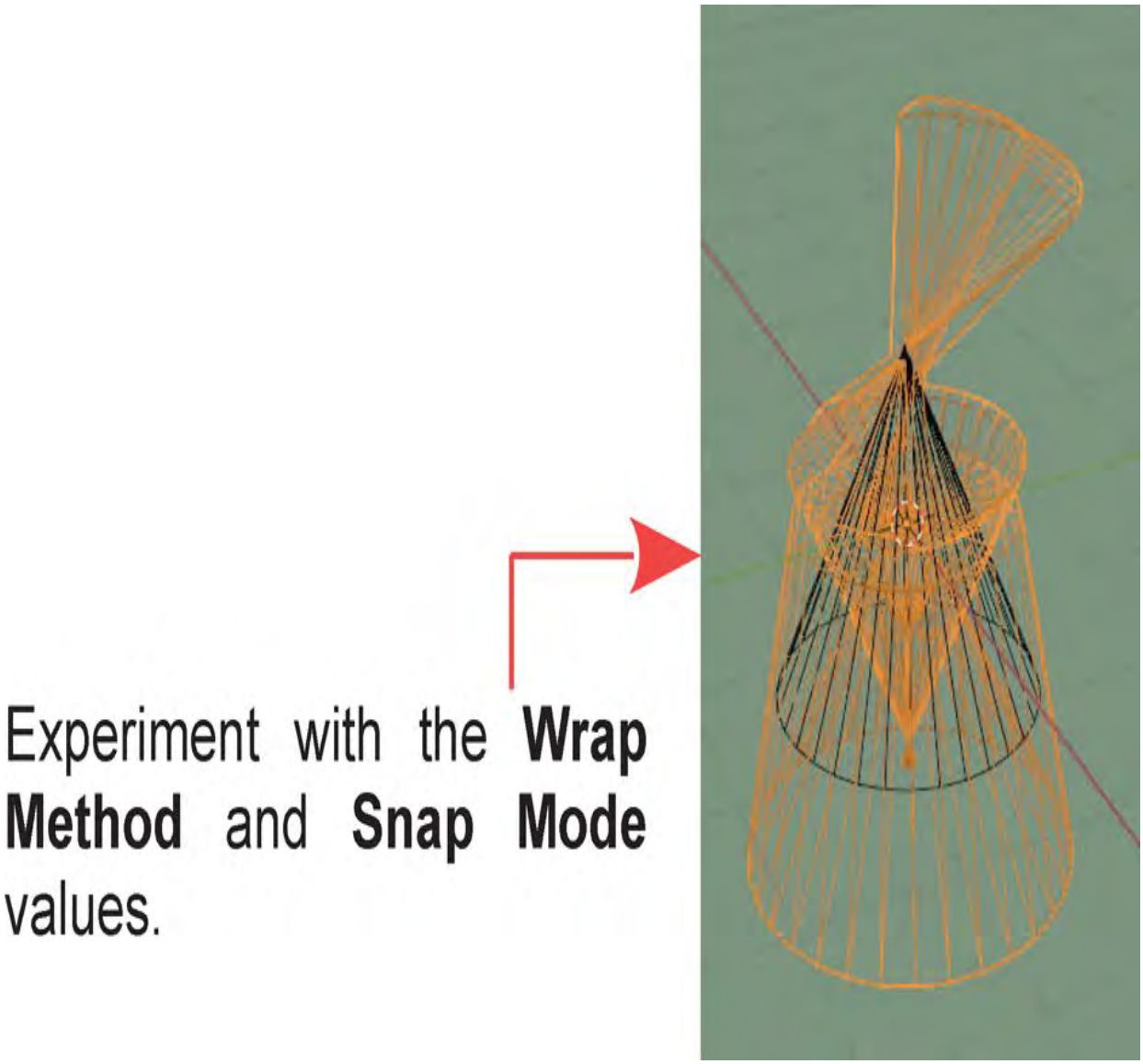


Figure 9.40

Altering the **Offset** value in the Modifier panel changes Sphere. the size of the modified UV

9.13 Simple Deform Modifier

The **Simple Deform Modifier** deforms a mesh by changing values in the Modifier and having a second Object with an influence. To see this Modifier in action, add a UV Sphere in the default Scene with a scaled down Cube located in the center of the Sphere (Figure 9.41). Activate **Wireframe**

Display Mode and add the **Simple Deform Modifier** to the UV Sphere with Cube entered as the **Origin** (Figure 9.42). When the Modifier is Added you immediately see the Sphere's mesh skew. **Note:** By default **Twist** is selected in the Modifier (highlighted blue).

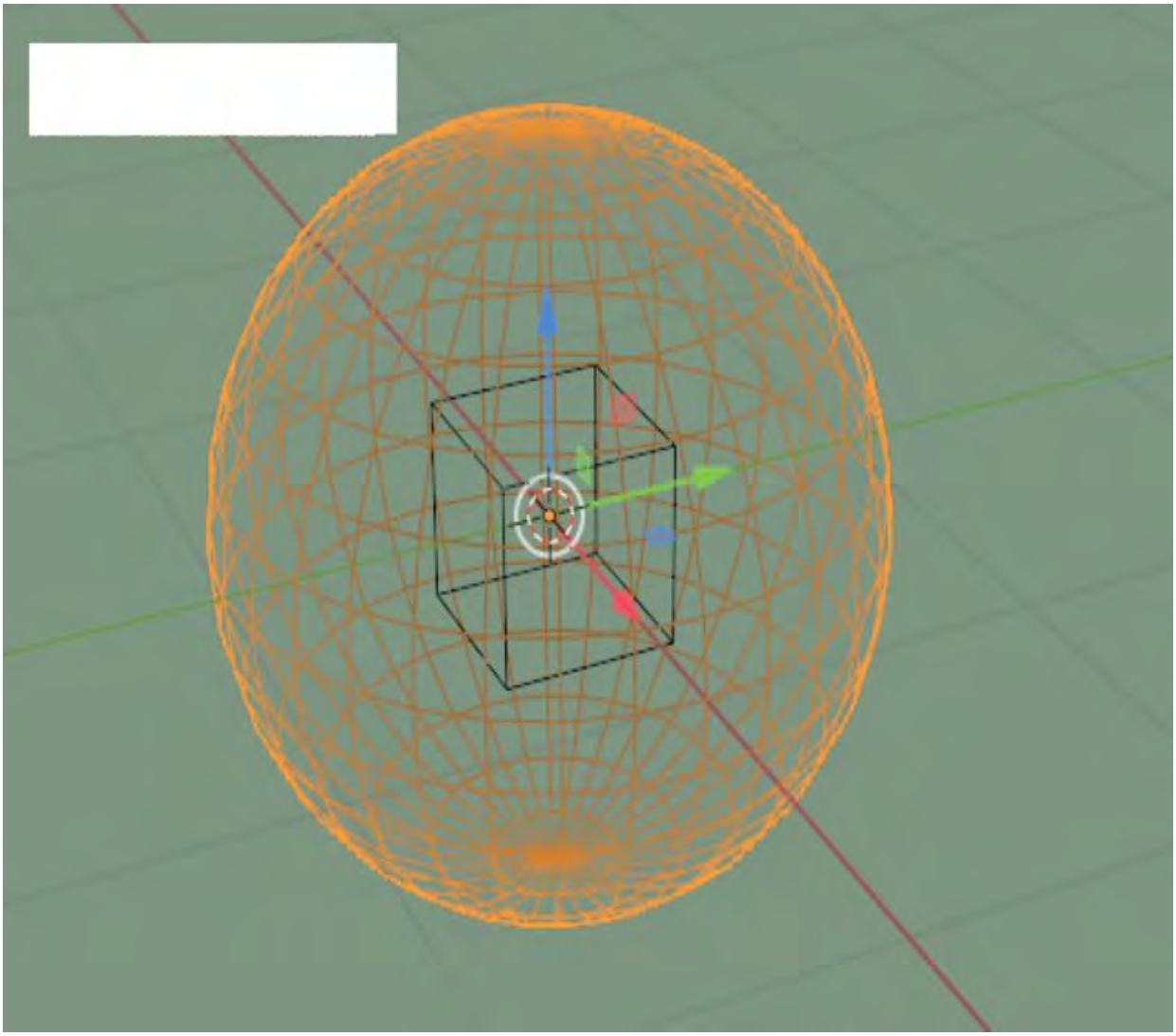


Figure 9.41

Note: By default **Twist** is selected in the Modifier (highlighted blue).

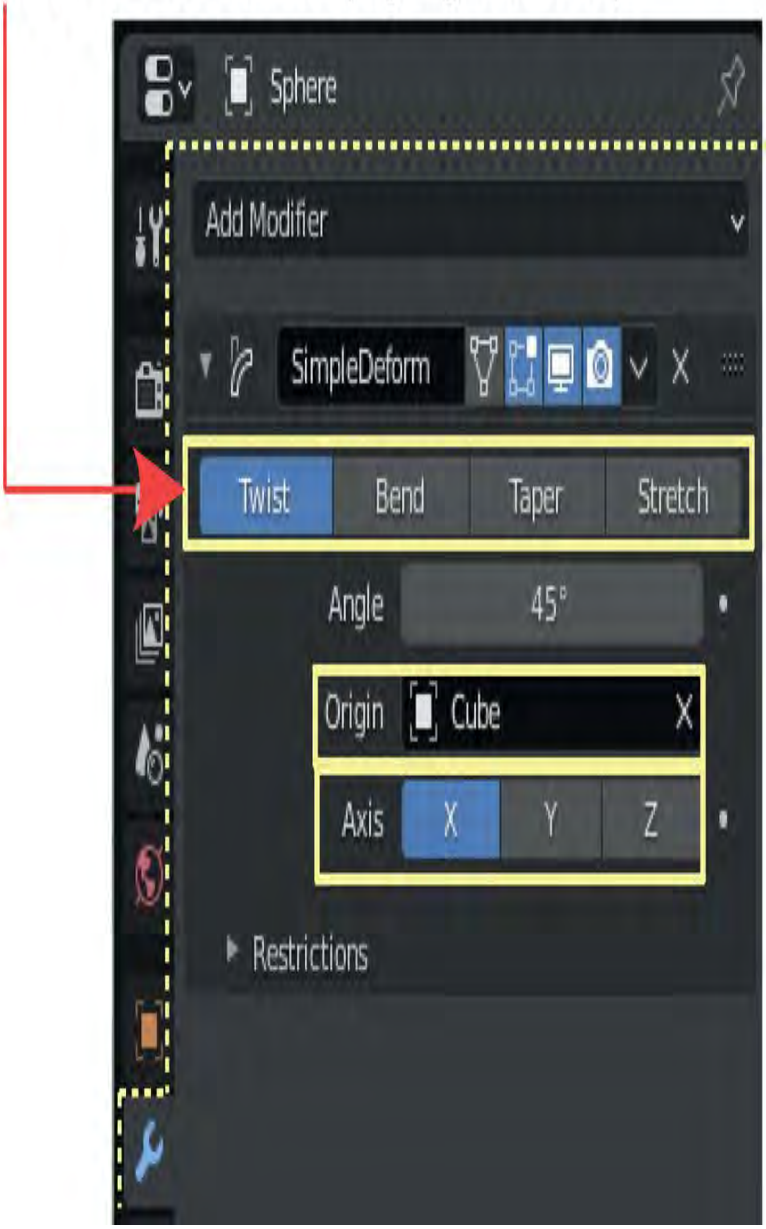
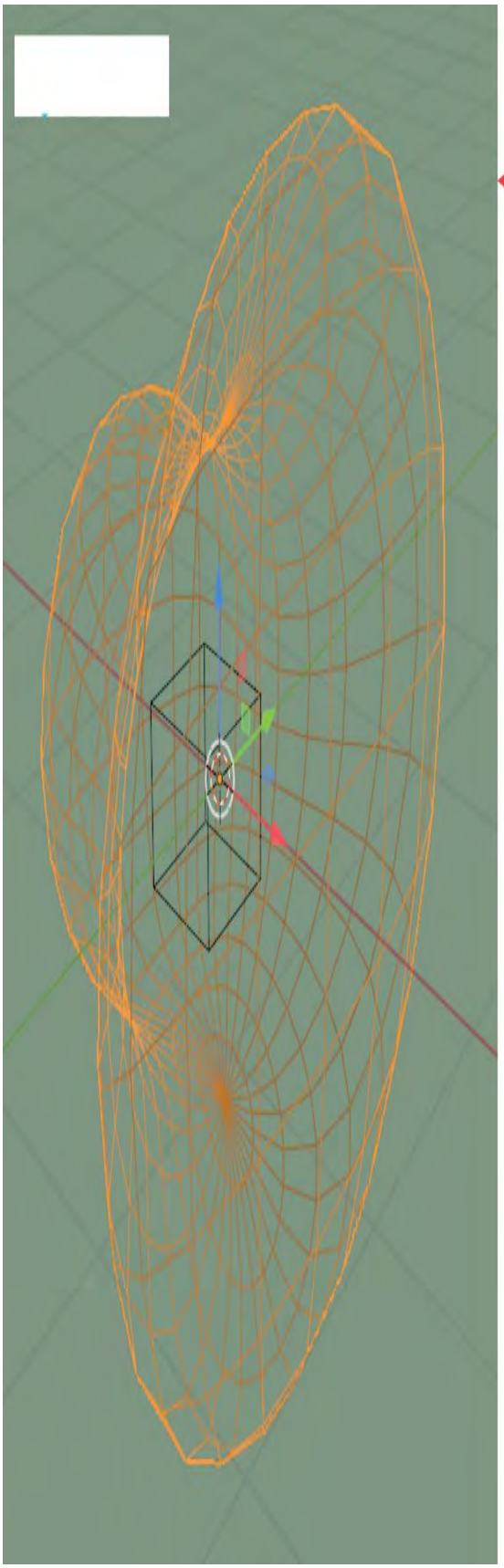


Figure 9.42

Try the the options: **Twist, Bend, Taper and Stretch.** Adjust the **Angle** slider to change the skew.

Experiment with the different settings in the Modifier panel. With any setting combination, Translating, Rotating or Scaling either the Cube or the UV Sphere affects the result. Deleting the Origin also affects results.



Stretch – Factor: 0.785 – Origin: Cube – Axis X

Figure 9.43

The result when the **Cube Origin** is removed.

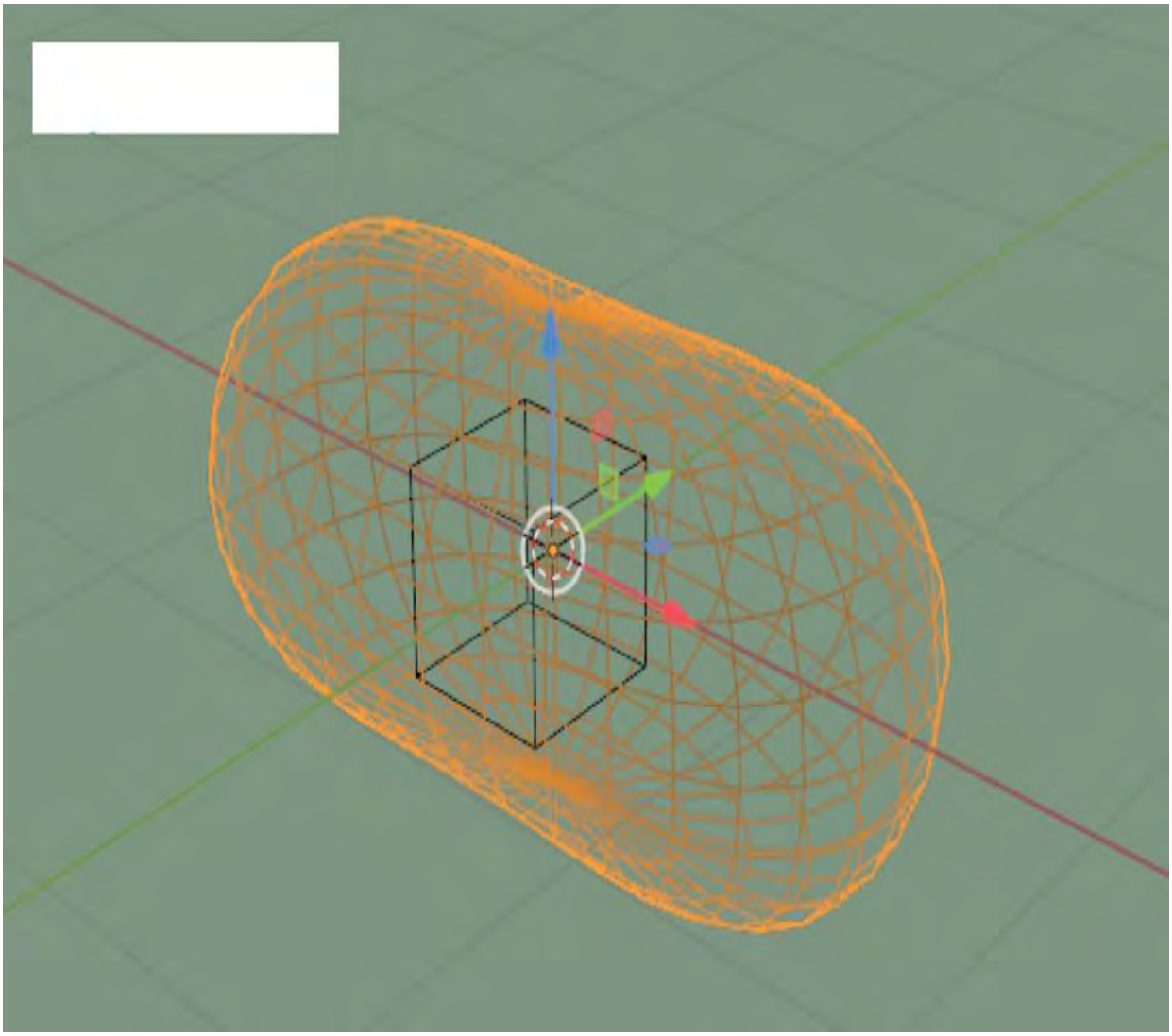


Figure 9.44

There are more settings in the Modifier, Restrictions Tab

9.14 Smooth Modifier

The **Smooth Modifier** smooths the mesh Object by softening the angles between adjacent Faces; this shrinks the size of the original Object at the same

time. **Note:** The smoothing effect is only applied to how the Object's surfaces are drawn in the 3D Viewport . In smoothing, no additional Vertices, Edges or Faces are added to the Object. To use the Smooth Modifier select an Object in the 3D Viewport then Add the Modifier in the **Properties Editor, Modifier Properties** (Figure 9.45).

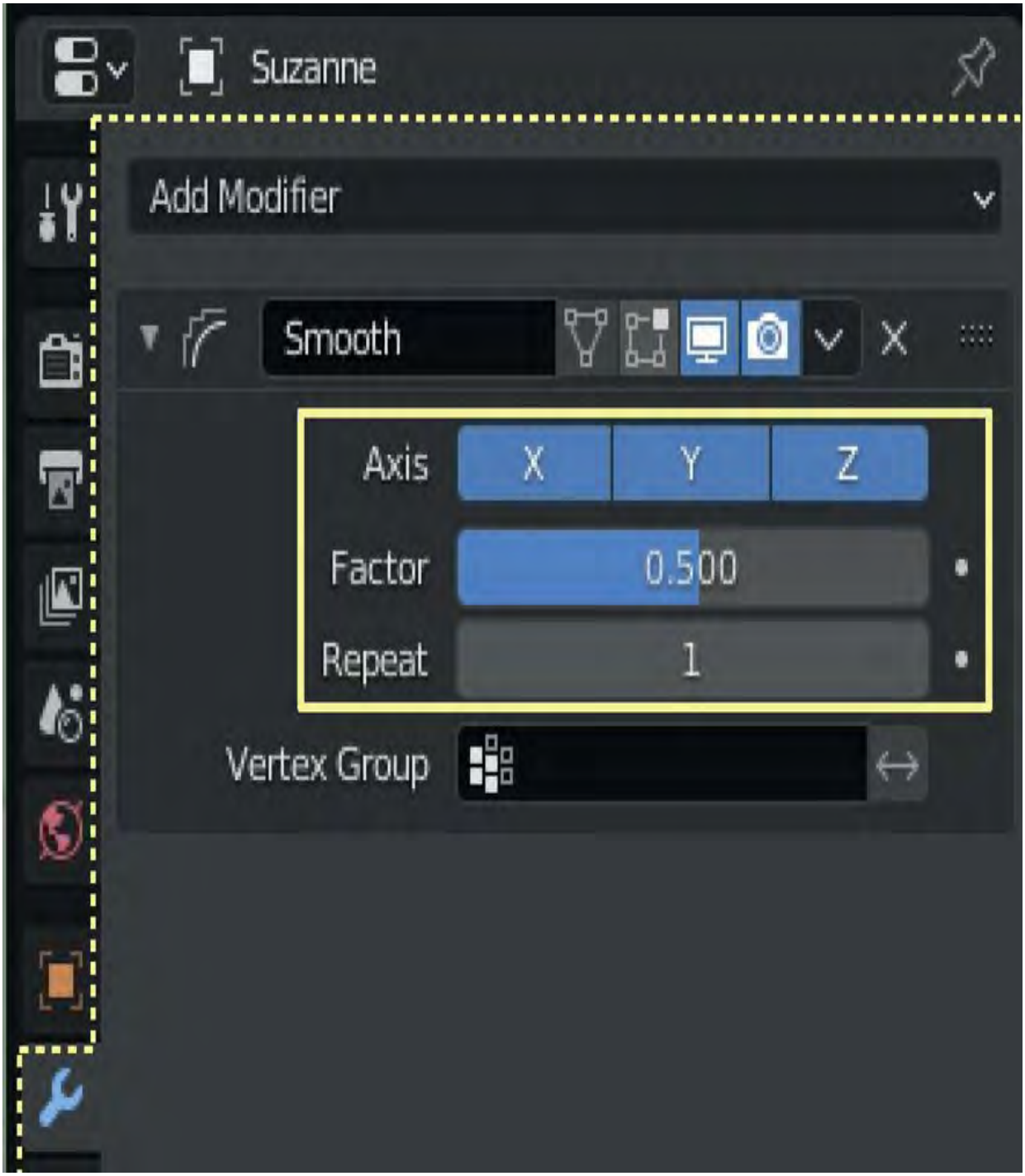


Figure 9.45

Adjust the **Factor slider** to increase or decrease smoothing. The **Repeat value** multiplies the Factor value. The **X,** Y and Z axis confine the smoothing to a particular axis.

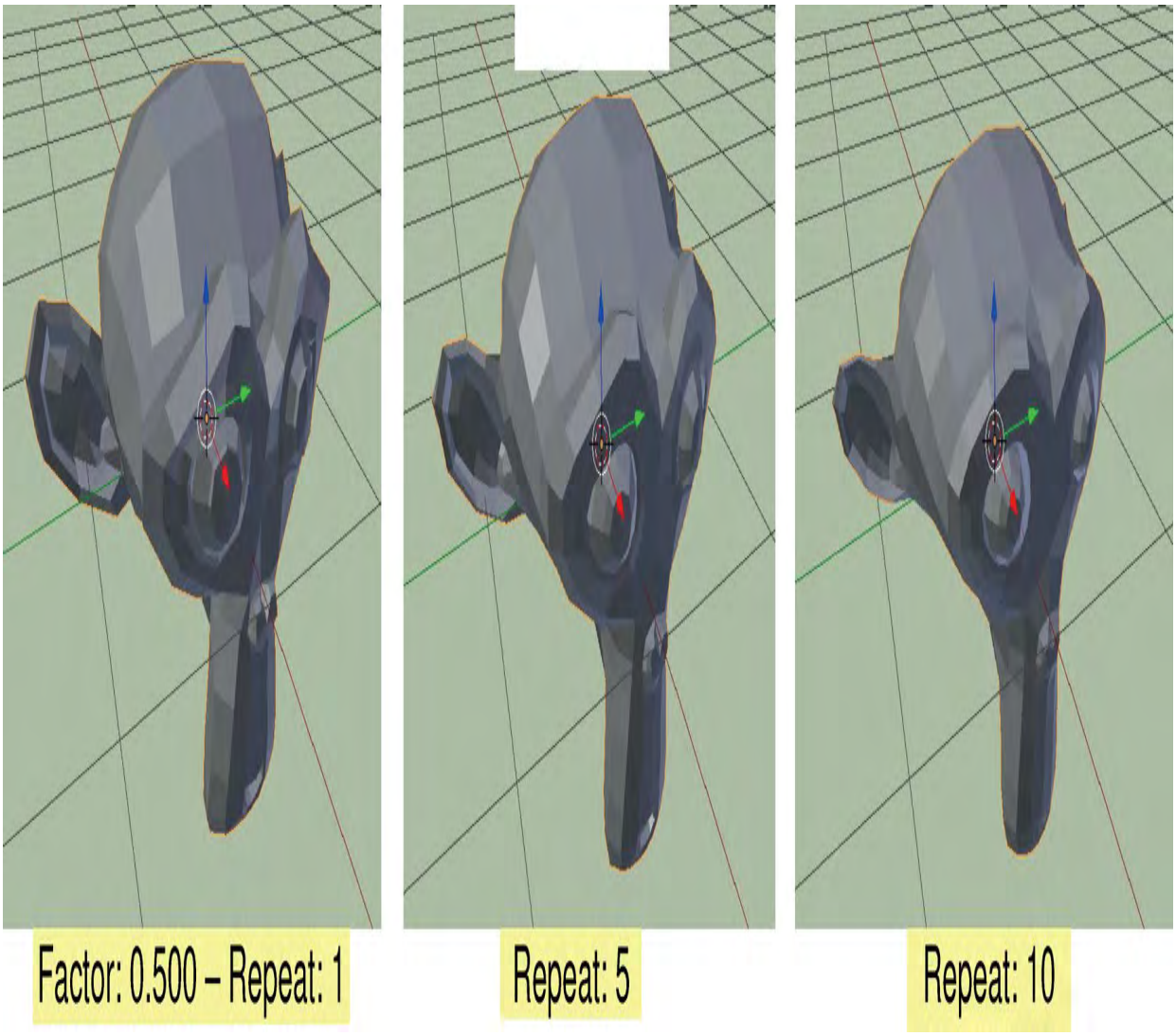


Figure 9.46

In applying the **Factor** and **Repeat** values the logic of the operation is to first set a value (the Factor) then perform the calculation (repeat) several times to achieve the desired smoothing. When the desired smoothing is achieved Apply the Modifier.

Note: An Object should have a reasonable number of Faces before the Modifier is effective. For example; the Monkey Object with its default Faces will shrink in size as **Repeats** are applied. When subdivided three or four times it will show a completely different effect.

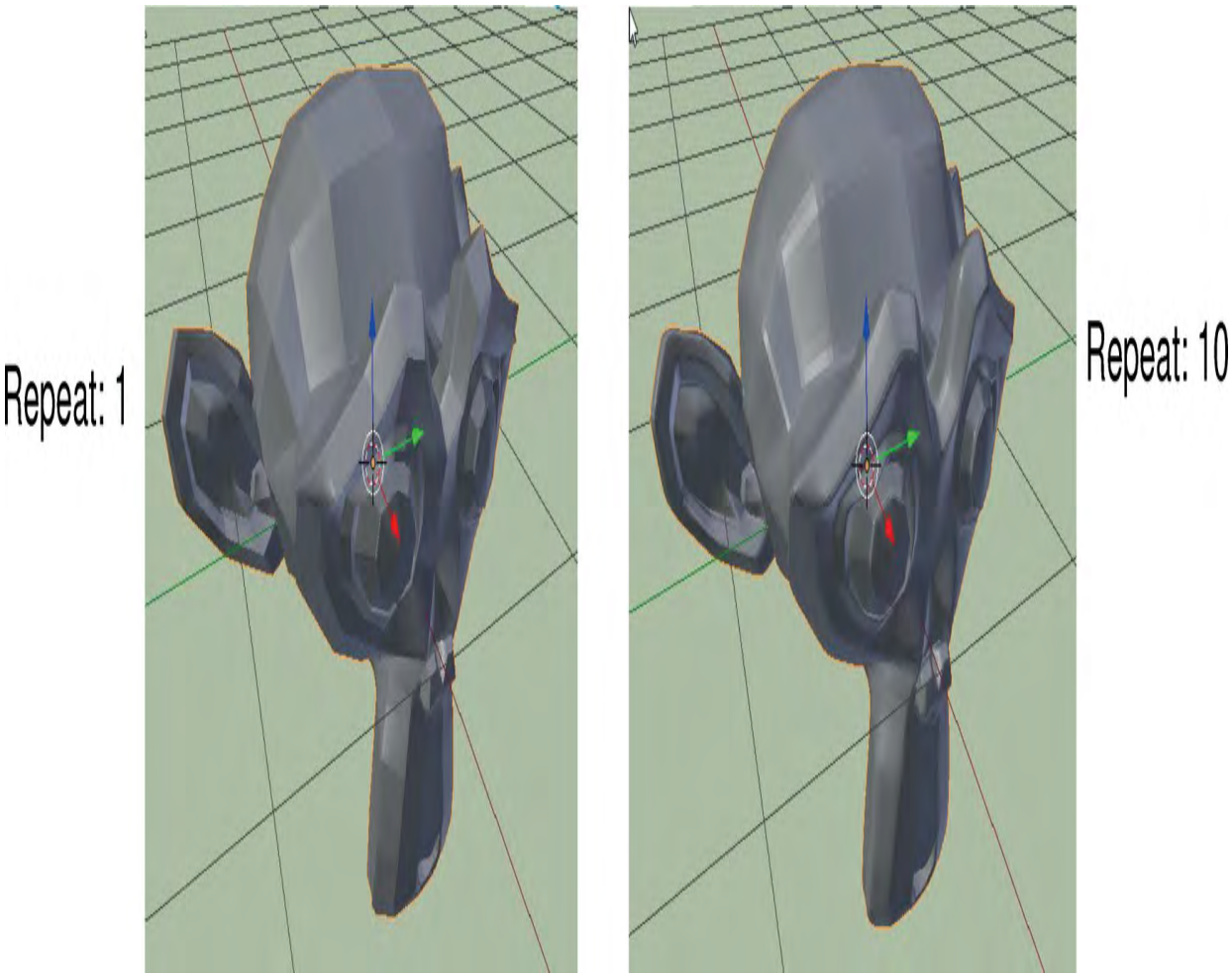


Figure 9.47

9.15 Smooth Corrective Modifier

The **Smooth Corrective Modifier** is used to Smooth and Correct imperfections in a model which occur when the surface is deformed. Figure 9.48 shows a model of an arm which has been posed and in doing so creates an imperfection (see Chapter 22).

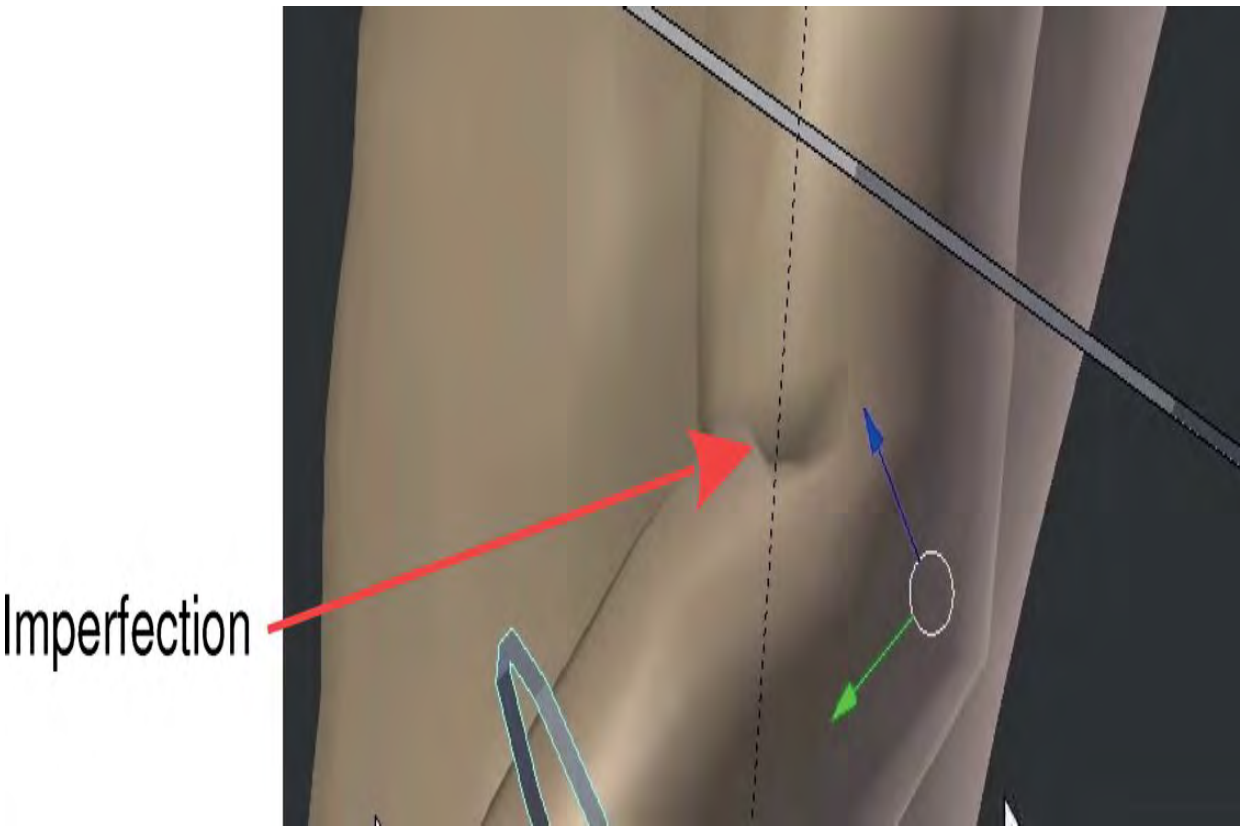


Figure 9.48

9.16 Smooth Laplacian Modifier

The **Laplacian Smooth Modifier** is used to smooth a mesh which has become irregular when Vertices have been manipulated during detailed modeling (Figures 9.49, 9.50).

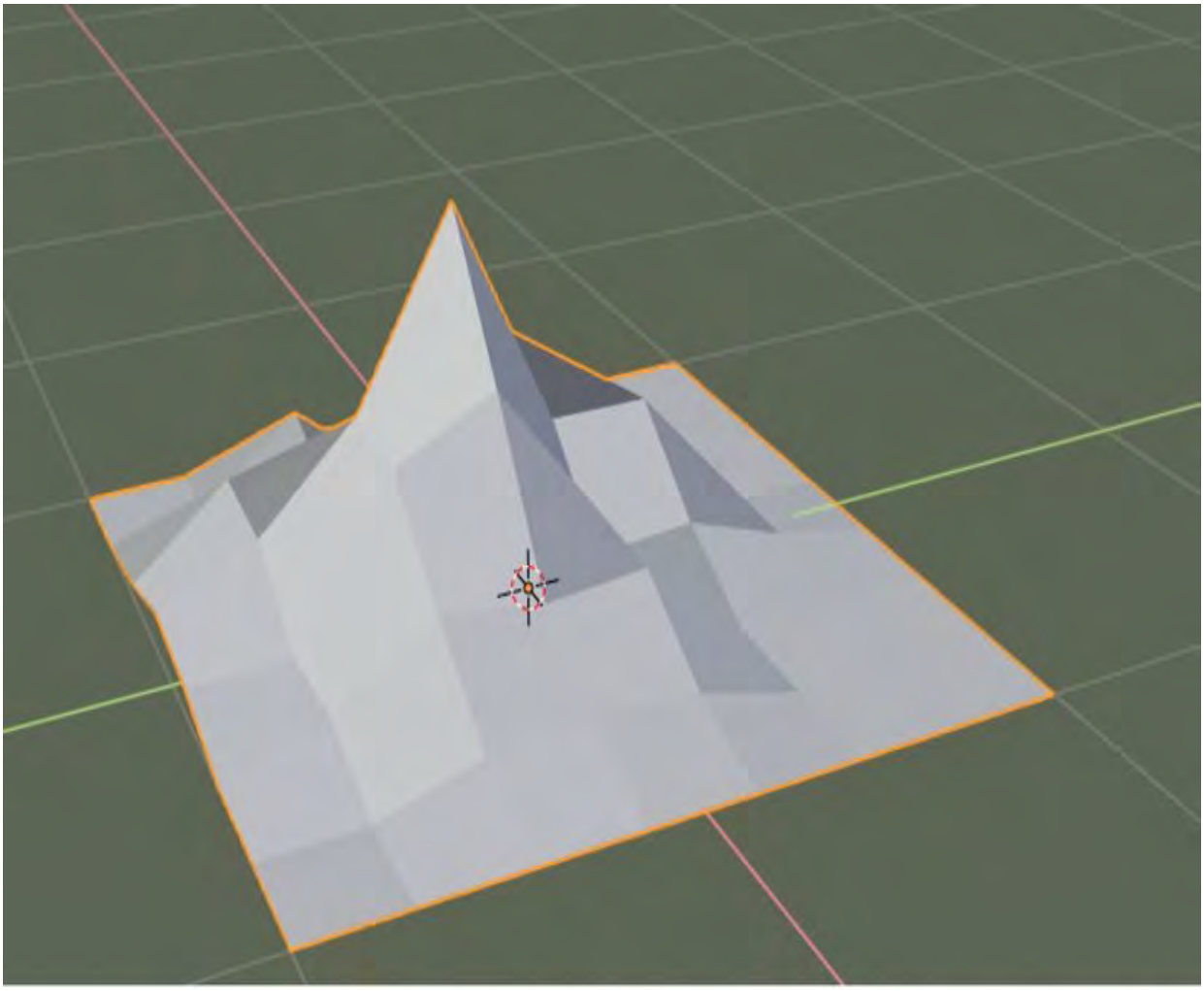


Figure 9.49

Plane Object, Subdivided — Number of Cuts: 7 One Vertex moved up with Proportional Editing enabled and Random Falloff.

With the Modifier added (NOT Applied) Edit Mode shows the Vertices in their original state (Figure 9.51).

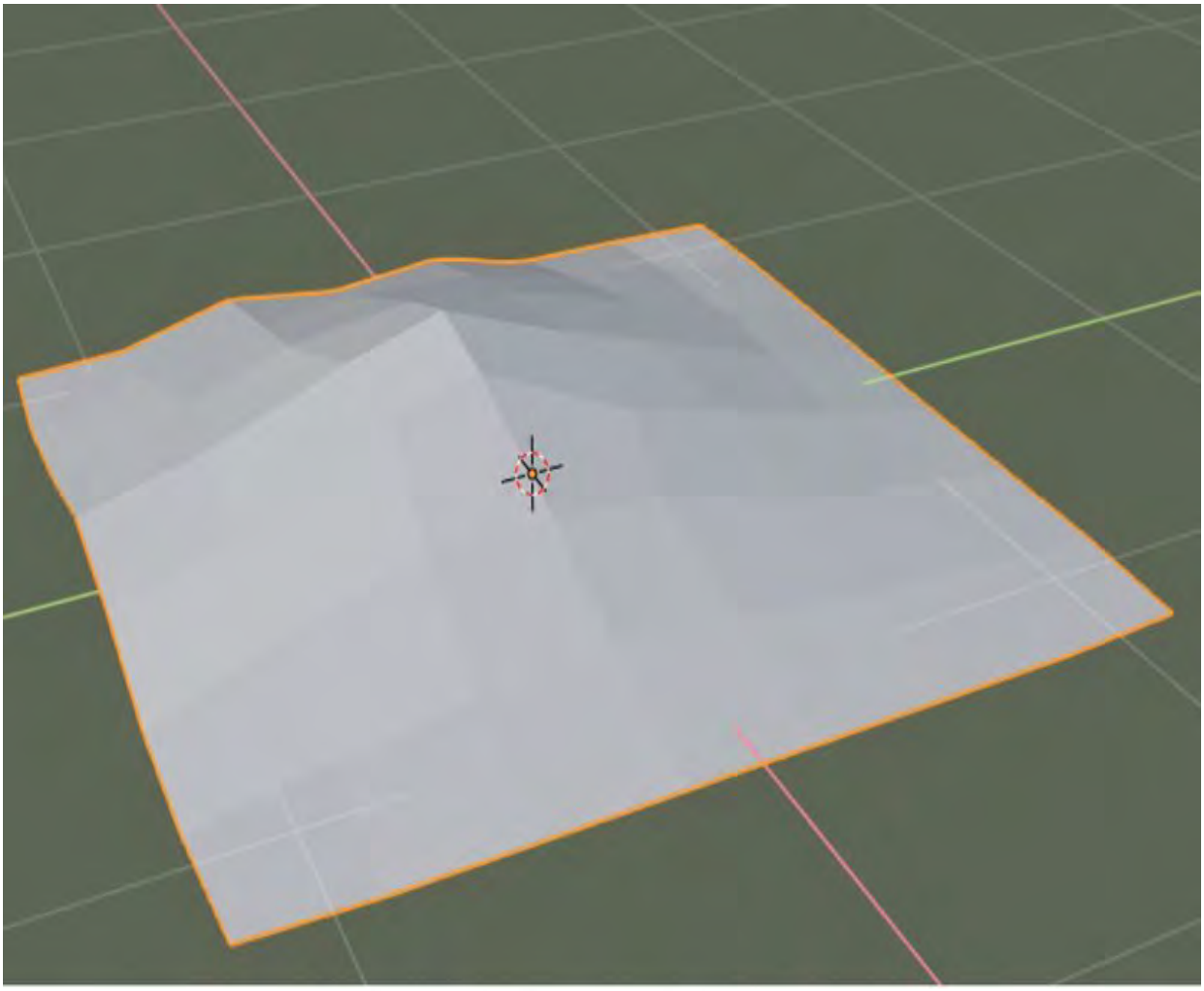


Figure 9.50

Smooth Laplacian Modifier added in Object Mode. Repeat Value: 5, Lambda Factor: 1.100.

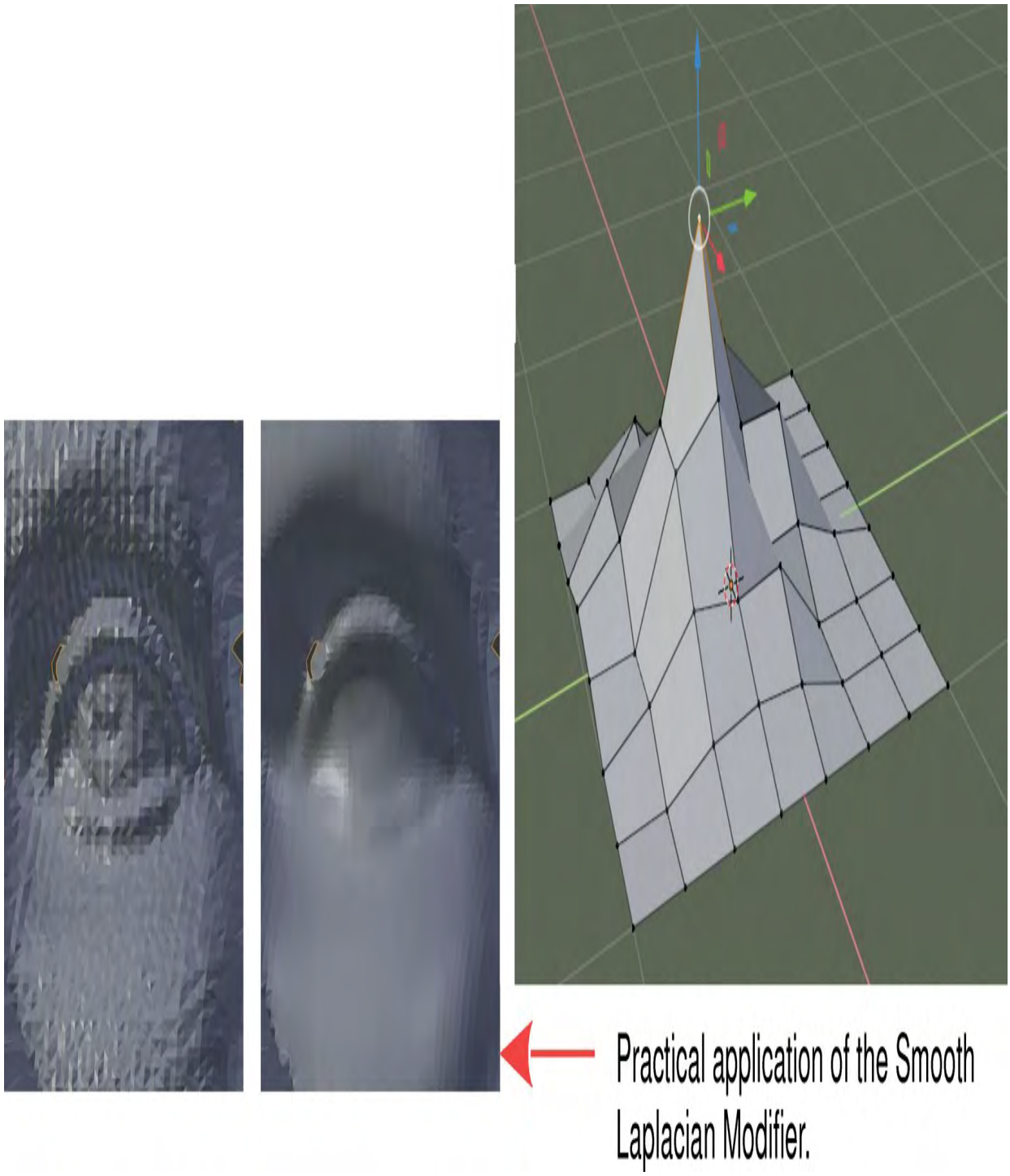


Figure 9.51

To permanently set the surface smooth, Apply the Modifier.

9.16 Surface Deform Modifier

The Surface Deform Modifier is similar to the Mesh Deform Modifier in that one mesh is used to shape another. The difference being, the controlling mesh does not have to surround the Object being shaped.

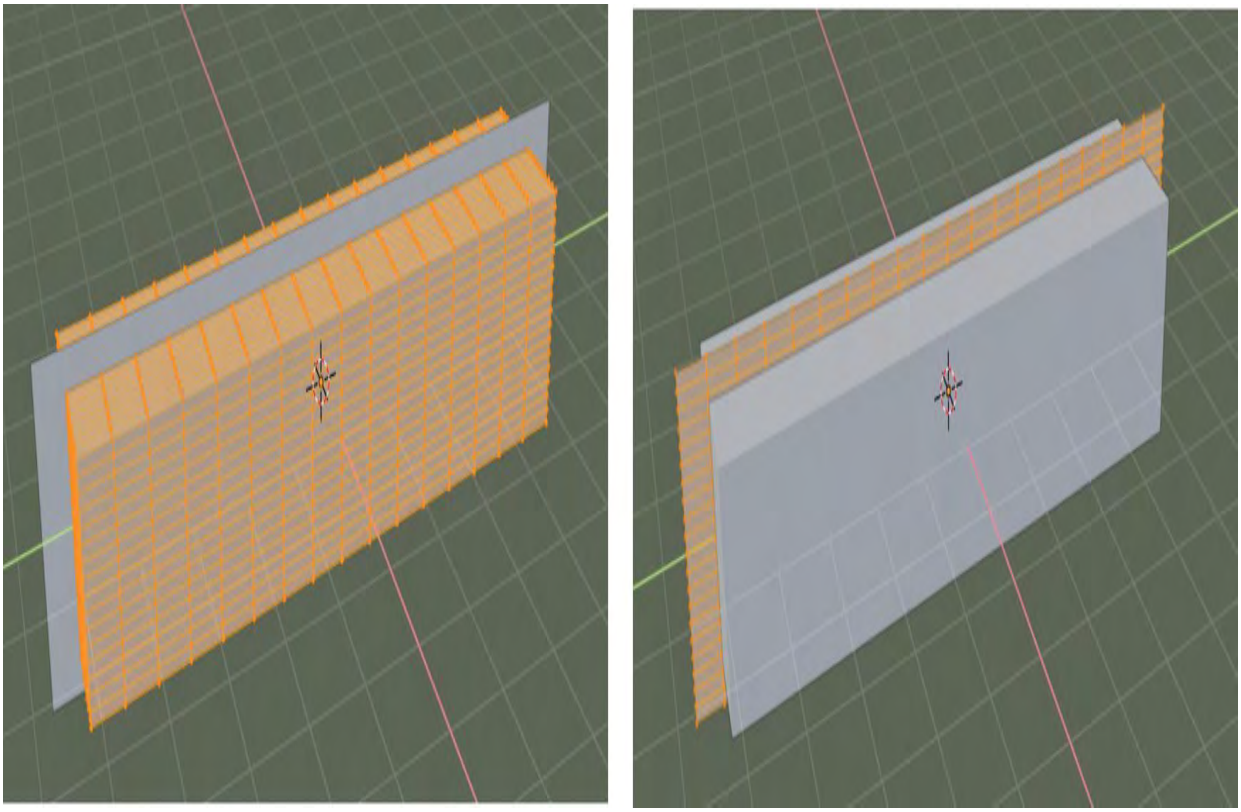


Figure 9.52

In Figure 9.52 a Cube Object has been Scaled along the Y Axis in Object Mode. In Edit Mode the Cube has been Subdivided with Number of Cuts: 10. A Plane Object has been added to the Scene, Rotated on the Y Axis and Scaled to bisect the elongated Cube. The Plane has also been Subdivided in Edit Mode, Number of Cuts: 10.

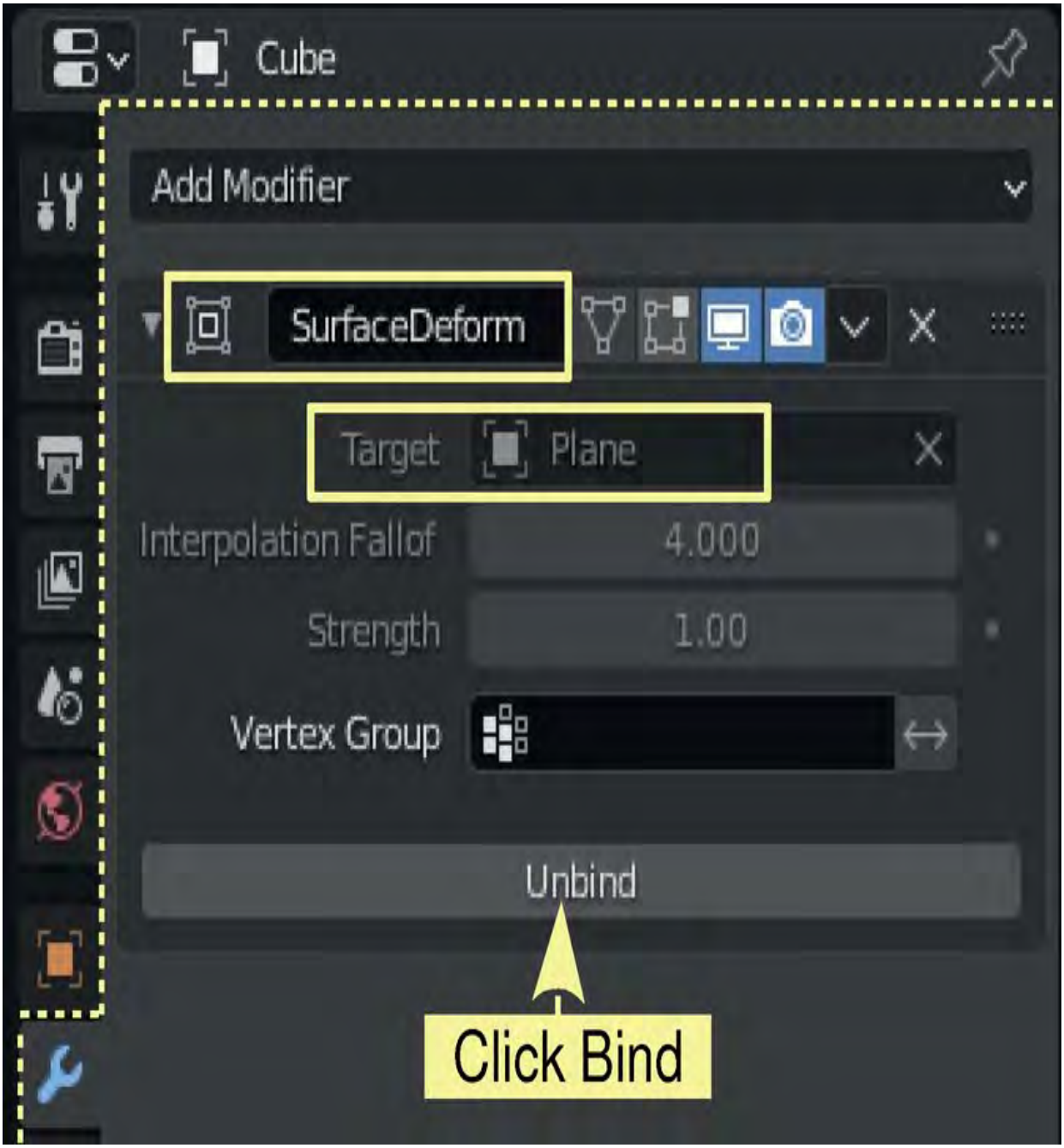


Figure 9.53

In Object Mode select the elongated Cube and add a **Surface Deform Modifier** (Figure 9.53). Enter **Plane** in the **Target Panel** as the controlling Object. **To set the Plane as the control click on Bind.**

Deforming the Plane will deform the elongated Cube (Figure 9.54 over).

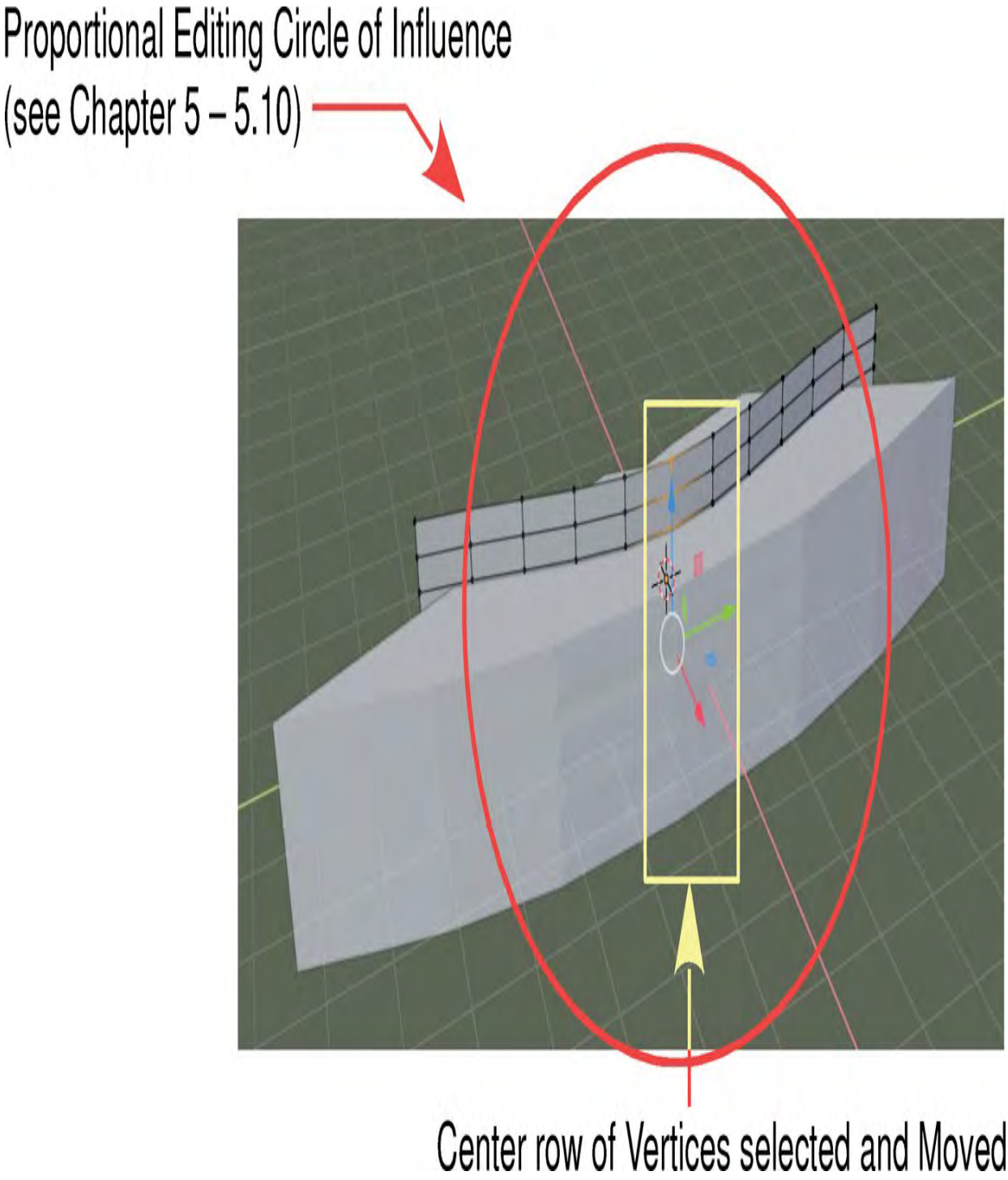


Figure 9.54

In Figure 9.54 the center row of Vertices of the Plane is selected (in Edit Mode) and Moved forward deforming the Cube. In performing this operation

Proportional Editing has been activated with Smooth Falloff and a Circle of Influence expanded to encapsulate the entire Plane.

After using this method to shape the Cube, Apply the Modifier and delete the Plane.

Animating the vertices of the Plane to move is an effective way of introducing lifelike characteristics to a Model. Figure 9.55 shows a Plane being animated to make the tail of a fish model move. (Reference: YouTube Video by yojigraphics)

https://www.youtube.com/watch?v=EJLIaNhoSSs

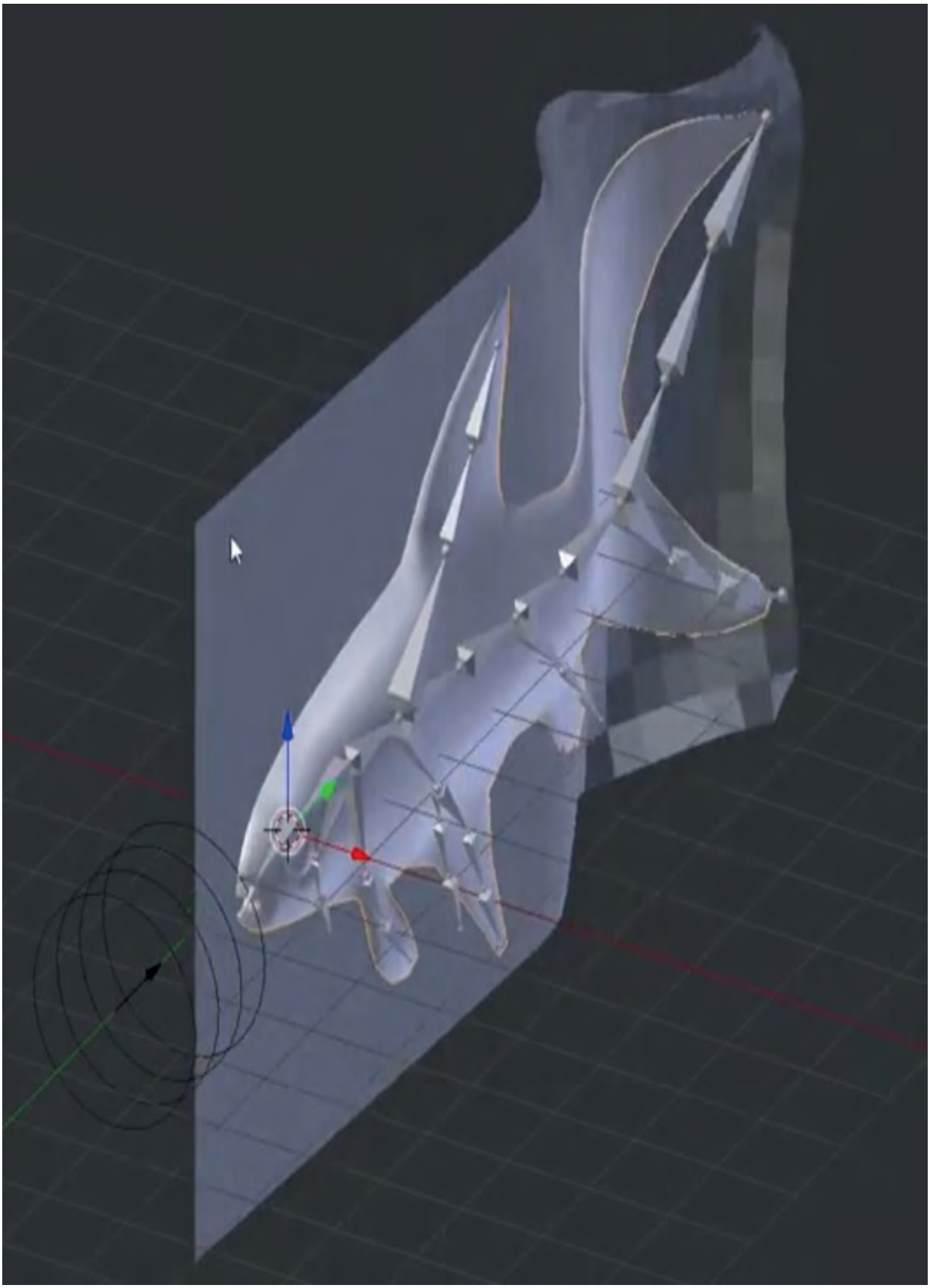


Figure 9.55

9.17 Warp Modifier

The **Warp Modifier** allows you to deform a mesh surface in Object mode by manipulating Target Objects. If you do not want the Targets to render in the Scene use Empty Objects. The deformation of the mesh takes place in a gradient between the two Targets.

This description requires clarification by a simple exercise.

In a new Blender Scene add a **Plane** Object, zoom in and **Subdivide** in Edit Mode (Cuts 10).

In Object Mode add two **Empty objects** and position as shown in Figure 9.56. Make note of the Empty names in the **Outliner Editor; Empty** and **Empty.001** (you may rename if you wish).

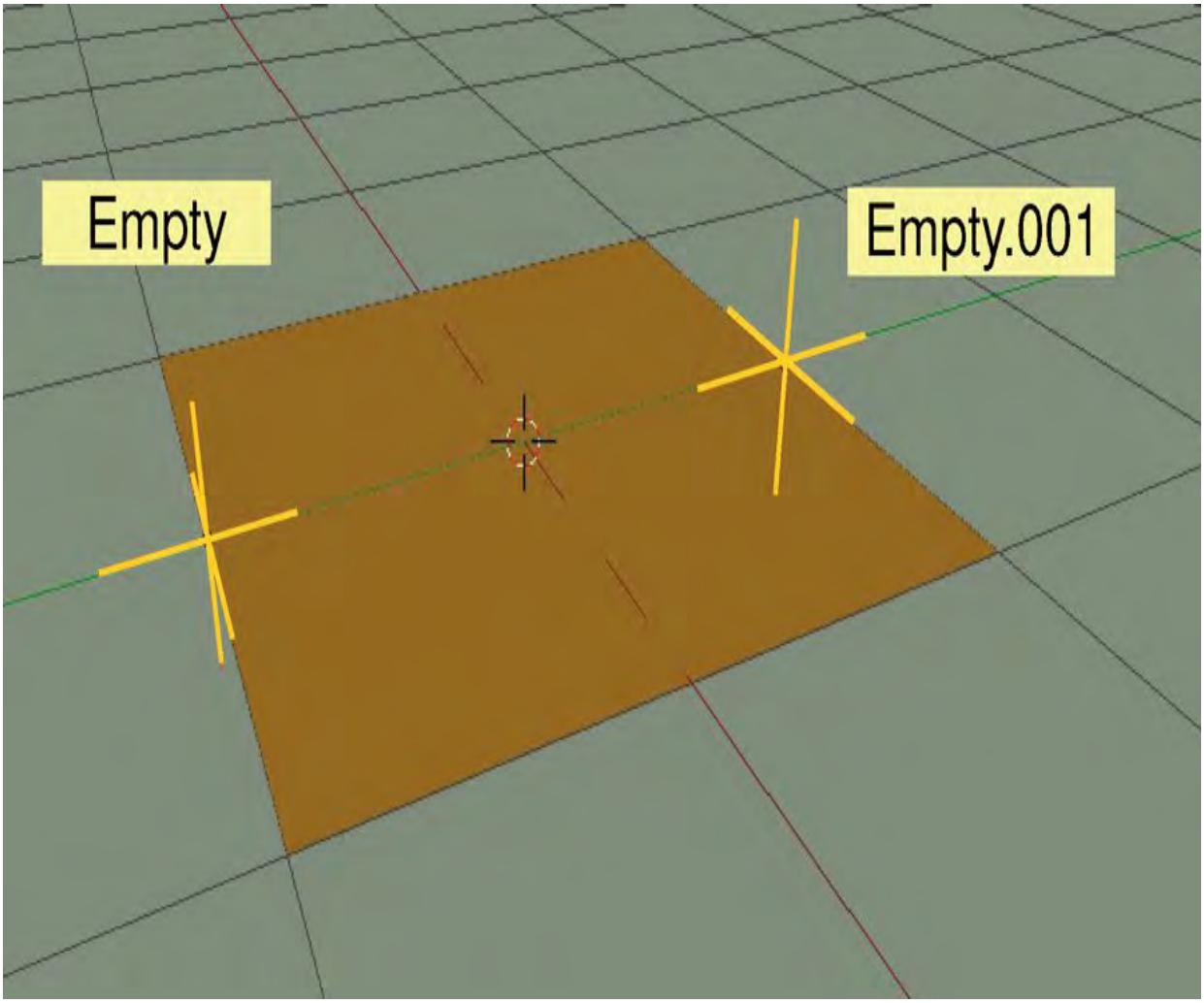


Figure 9.56

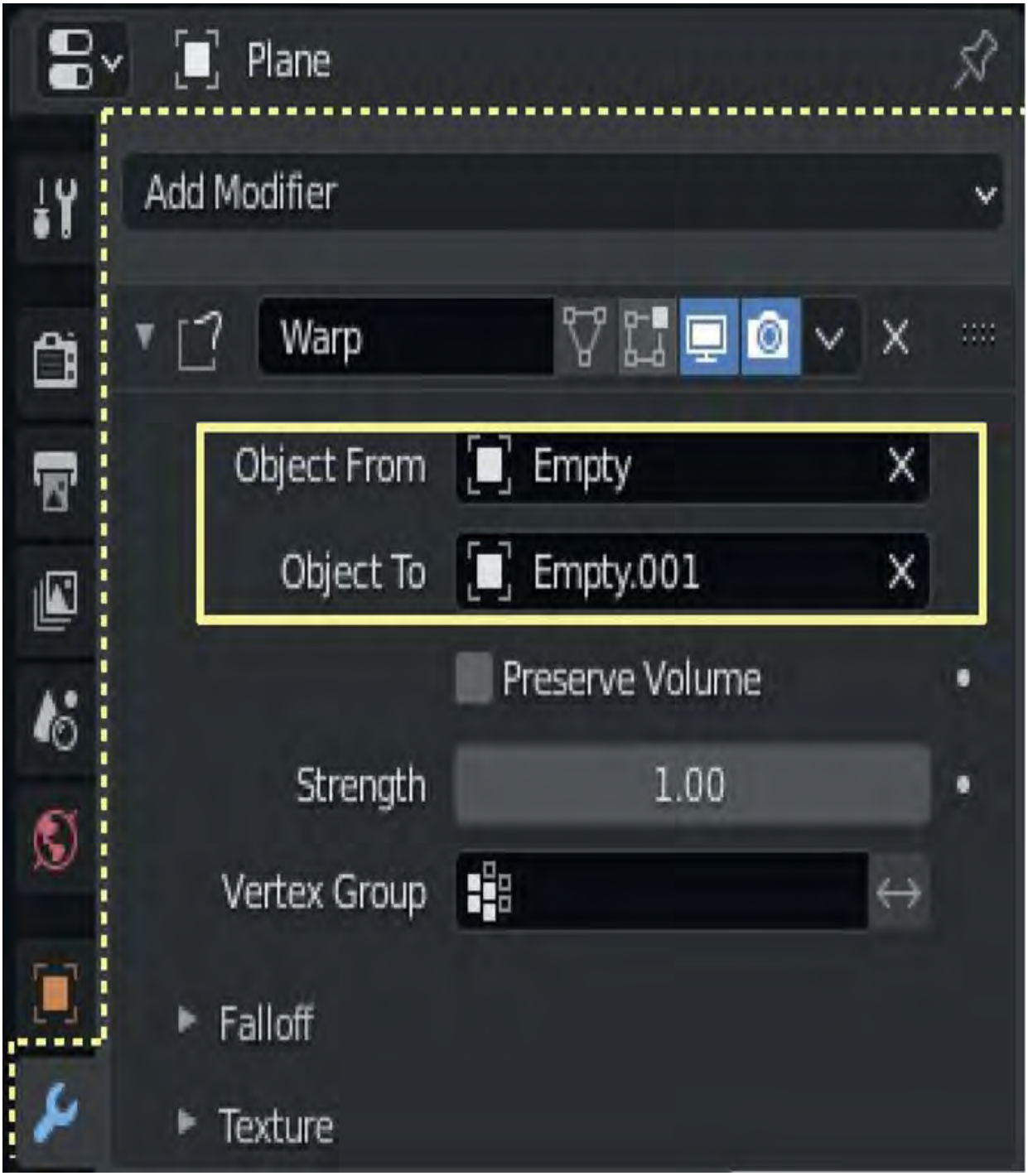


Figure 9.57

Select the Plane and in the Properties Editor, Modifier buttons add a **Warp Modifier** (Figure 9.57).

In the Modifier panel enter the names of the Empty Target Objects in the **From** and **To** panels (From_ Empty and To: Empty.001). You will immediately see the Plane deform in the 3D View Editor (Figure 9.58).

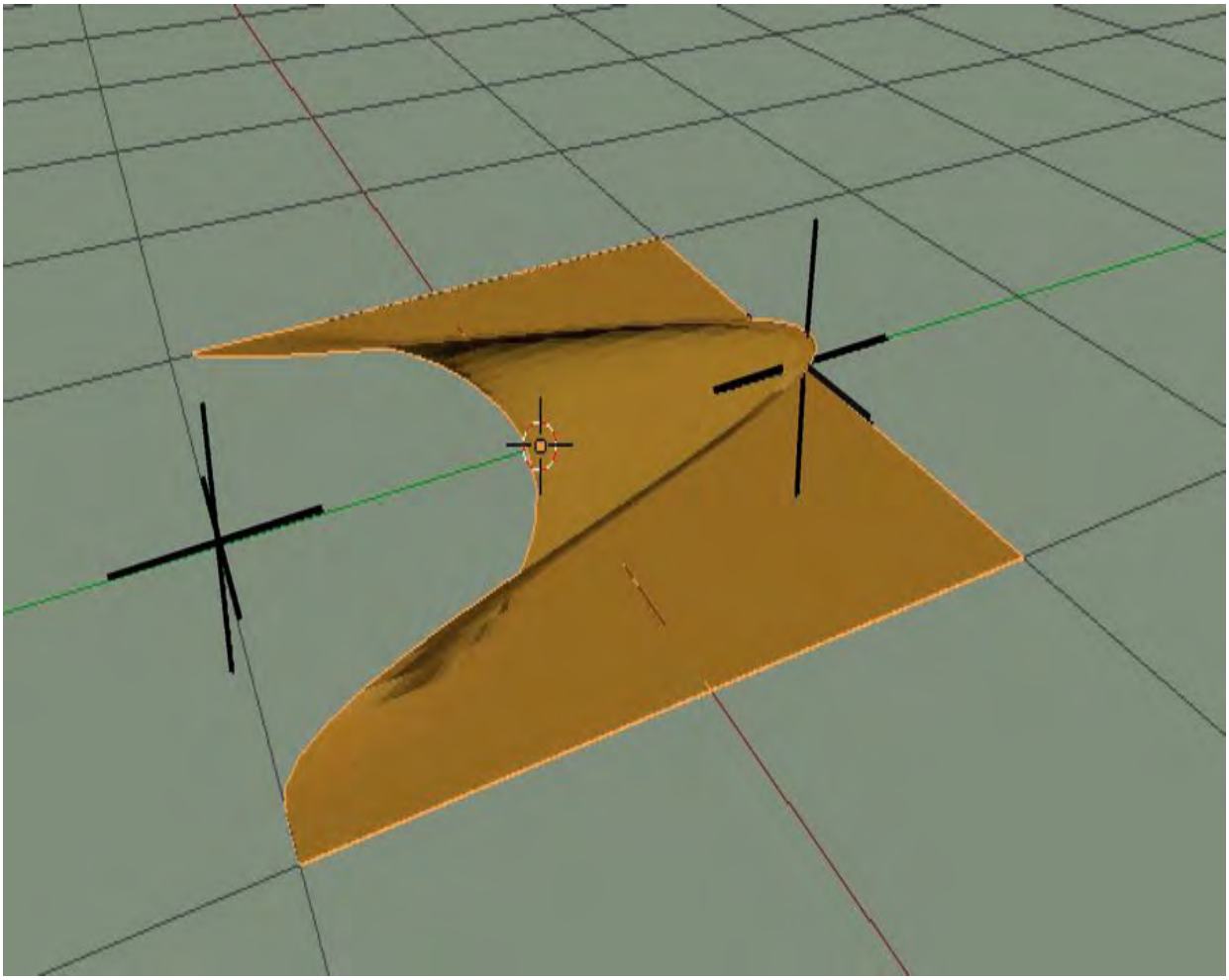


Figure 9.58

You may consider the Modifier as saying; deform the mesh From Empty to Empty.001.

By selecting either Empty in the 3D View Editor and translating the deformation of the Plane is affected (Figure 9.59).

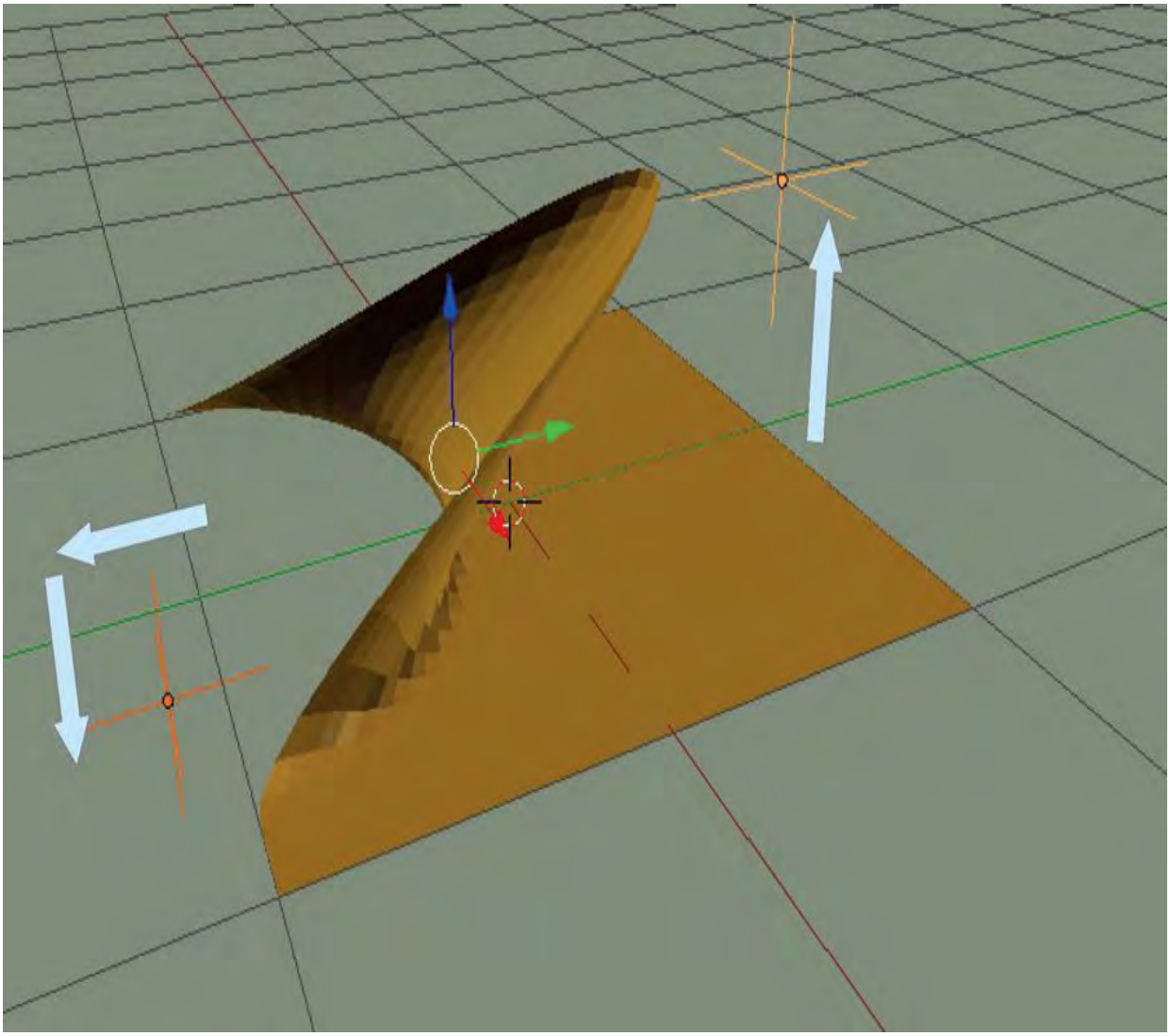


Figure 9.59

In the Modifier panel you may choose a different **Falloff Type** by opening the Falloff Tab (Figure 9.57). Adjust the **Strength** and **Radius** sliders to modify the deformation (Figure 9.60).

When the Modifier is Applied the shape of the Plane is permanently set.

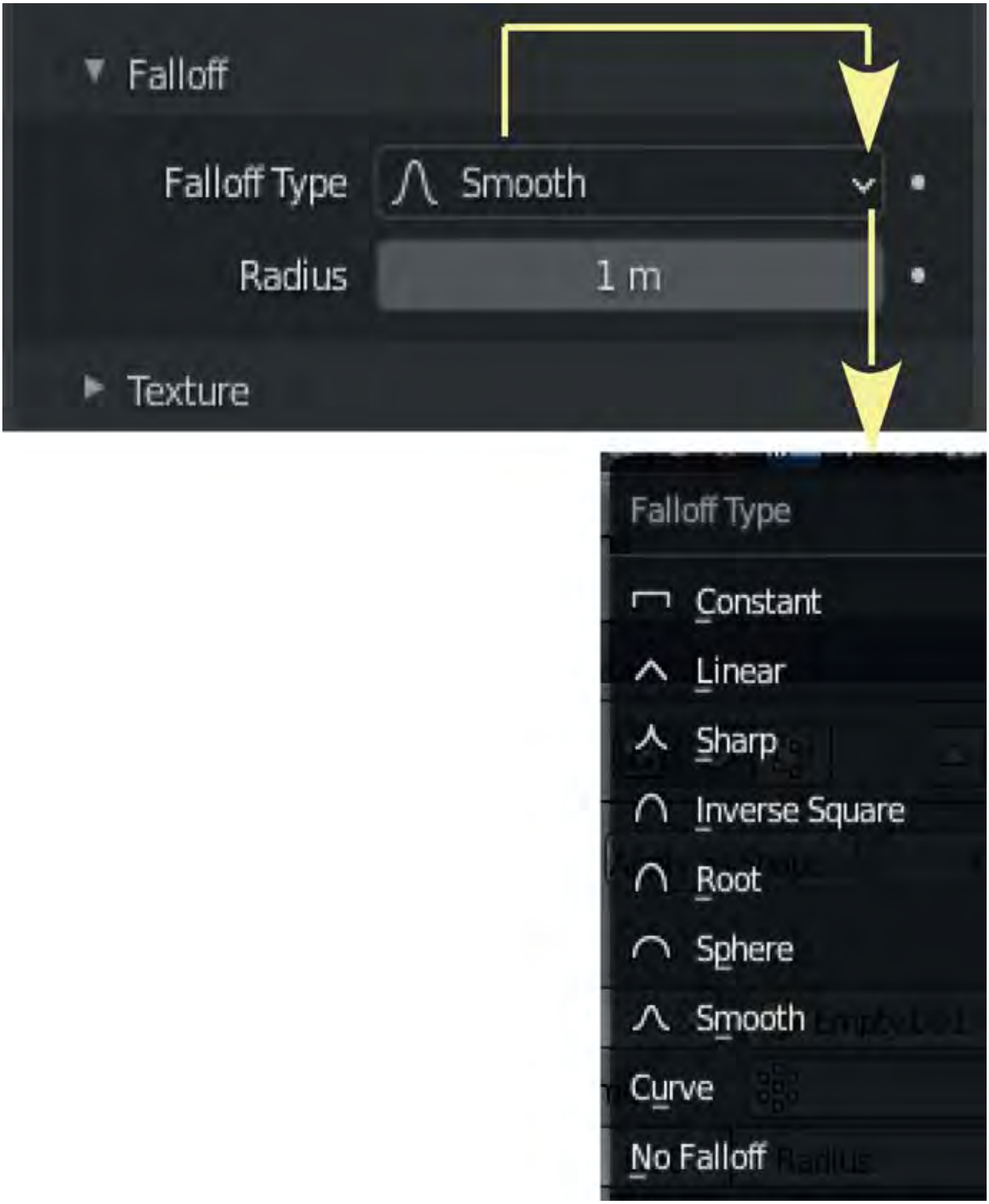

Falloff
Falloff Type
Smooth
Radius
1 m
Texture
Falloff Type
Constant
Linear
Sharp
Inverse Square
Root
Sphere
Smooth
Curve
No Falloff

Figure 9.60

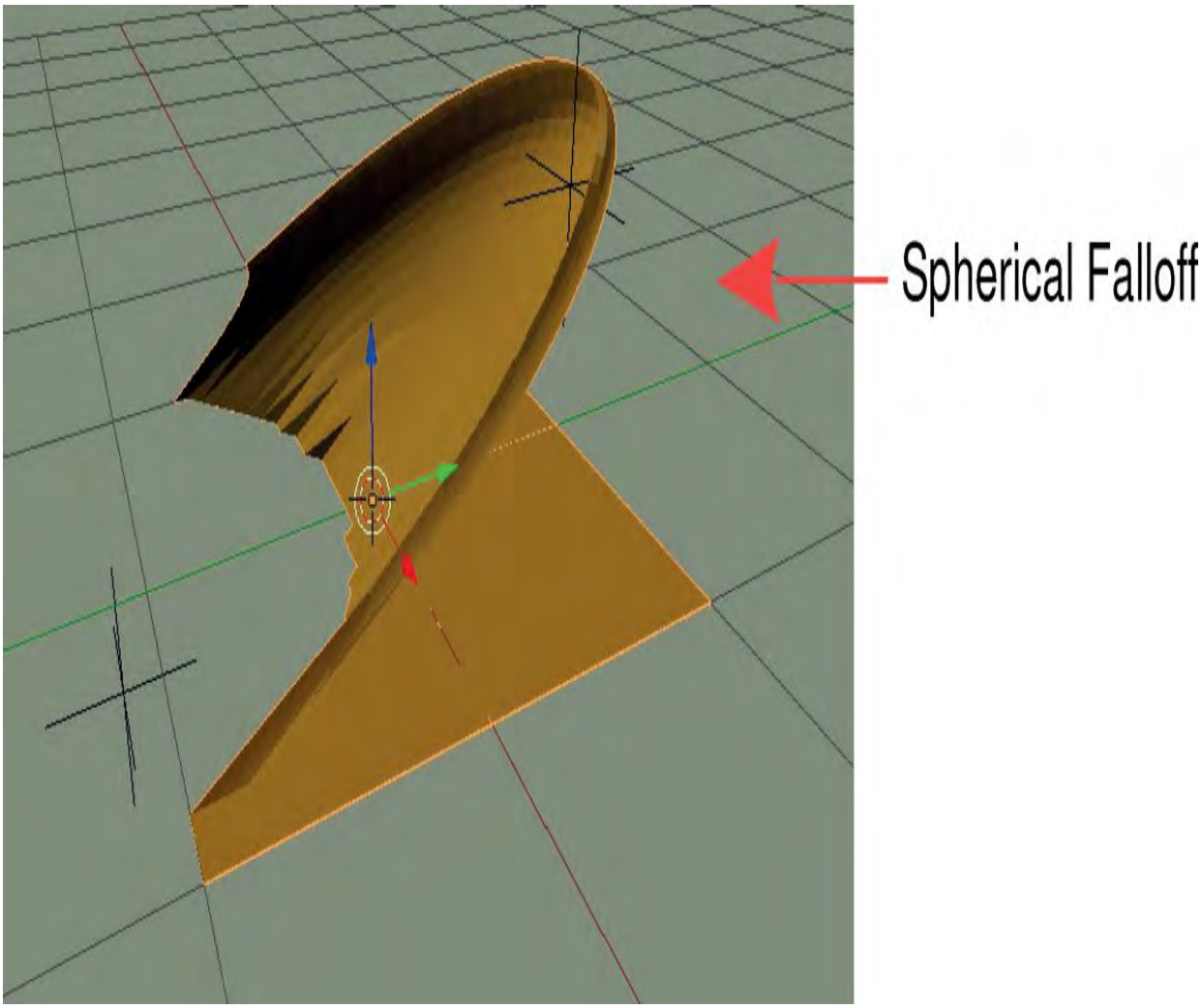


Figure 9.61

Note: In Figures 9.61 and 9.62 the position of the Empty target objects have NOT been moved.

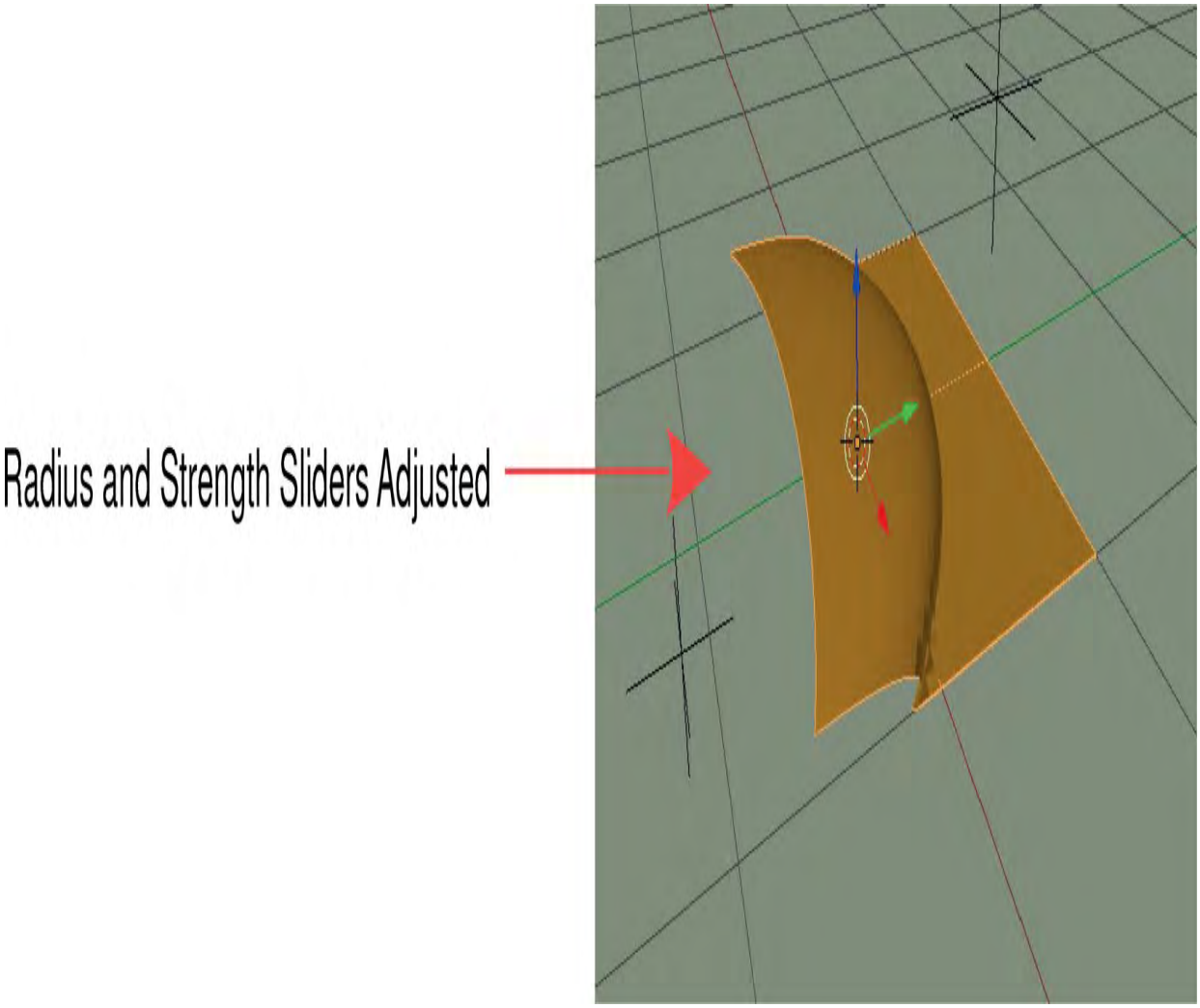


Figure 9.62

9.18 Wave Modifier

The **Wave Modifier** applies a deformation and creates an animation in a wave form. To demonstrate, in the default Blender Scene, delete the Cube and add a **Plane.** Scale the Plane up six times, Tab into **Edit mode,** and Subdivide the Plane by clicking **RMB - Subdivide** in the **Mesh Context** menu. In the **Subdivide** panel, make **Number of Cuts** 15. **Tab** back to **Object Mode.**

Plane
Add Modifier
Wave
Motion
X
Y
Cyclic
Along Normals
X
Y
Z
Falloff
0 m
Height
0.5 m
Width
1.5 m
Narrowness
1.5 m
Vertex Group
Start Position
Time
Texture

Figure 9.63

With the Plane selected, in the **Properties Editor, Modifiers Properties** add a **Wave Modifier** (Figure 9.63).

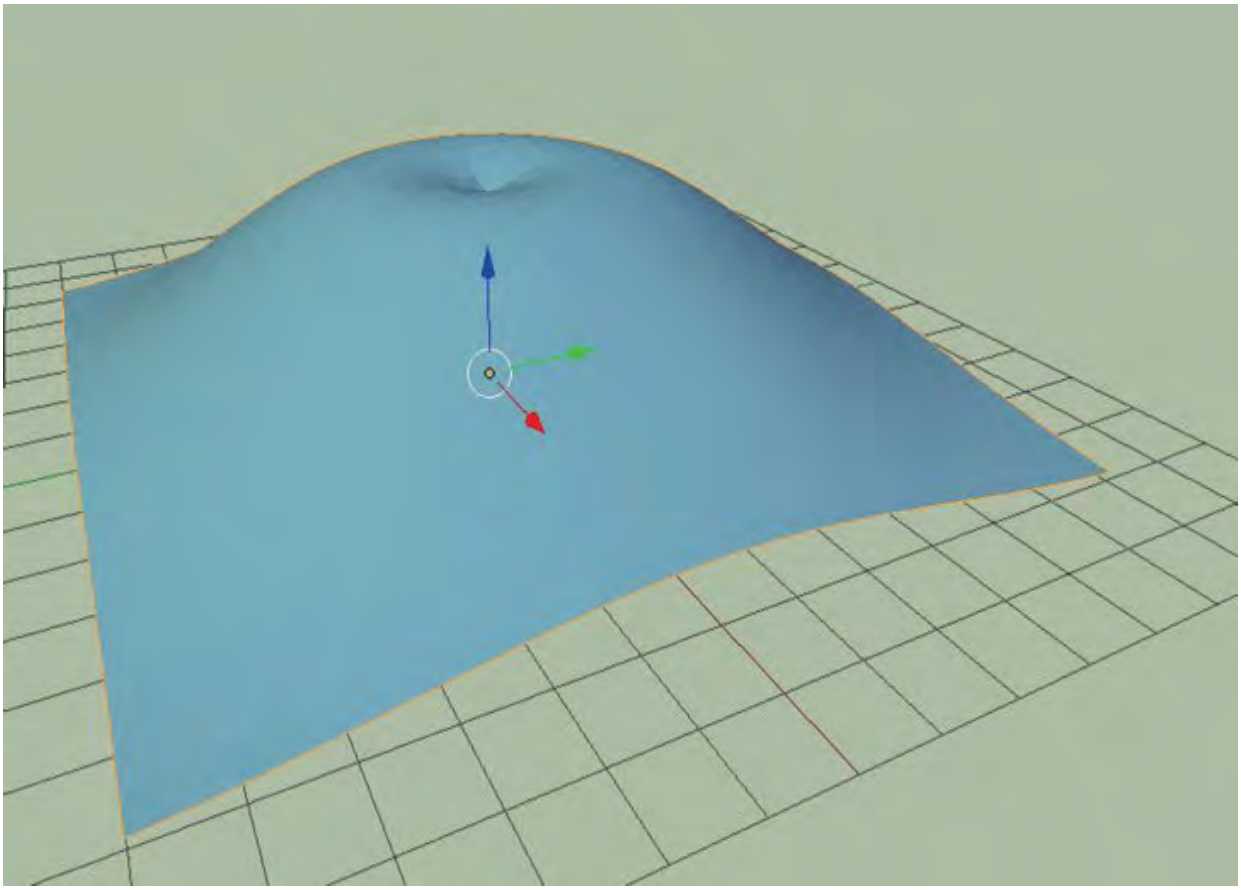


Figure 9.64

You immediately see the plane deform in the 3D Viewport Editor, pulled up in the middle and punched in at the top of the bulge (Figure 9.65).

The Wave Modifier has been applied on both the **X** and Y axis. In the Modifier panel you see **X,** Y, and Cyclic selected. **X** and Y refer to the axis and **Cyclic** means that an animation of the wave will repeat over and over.

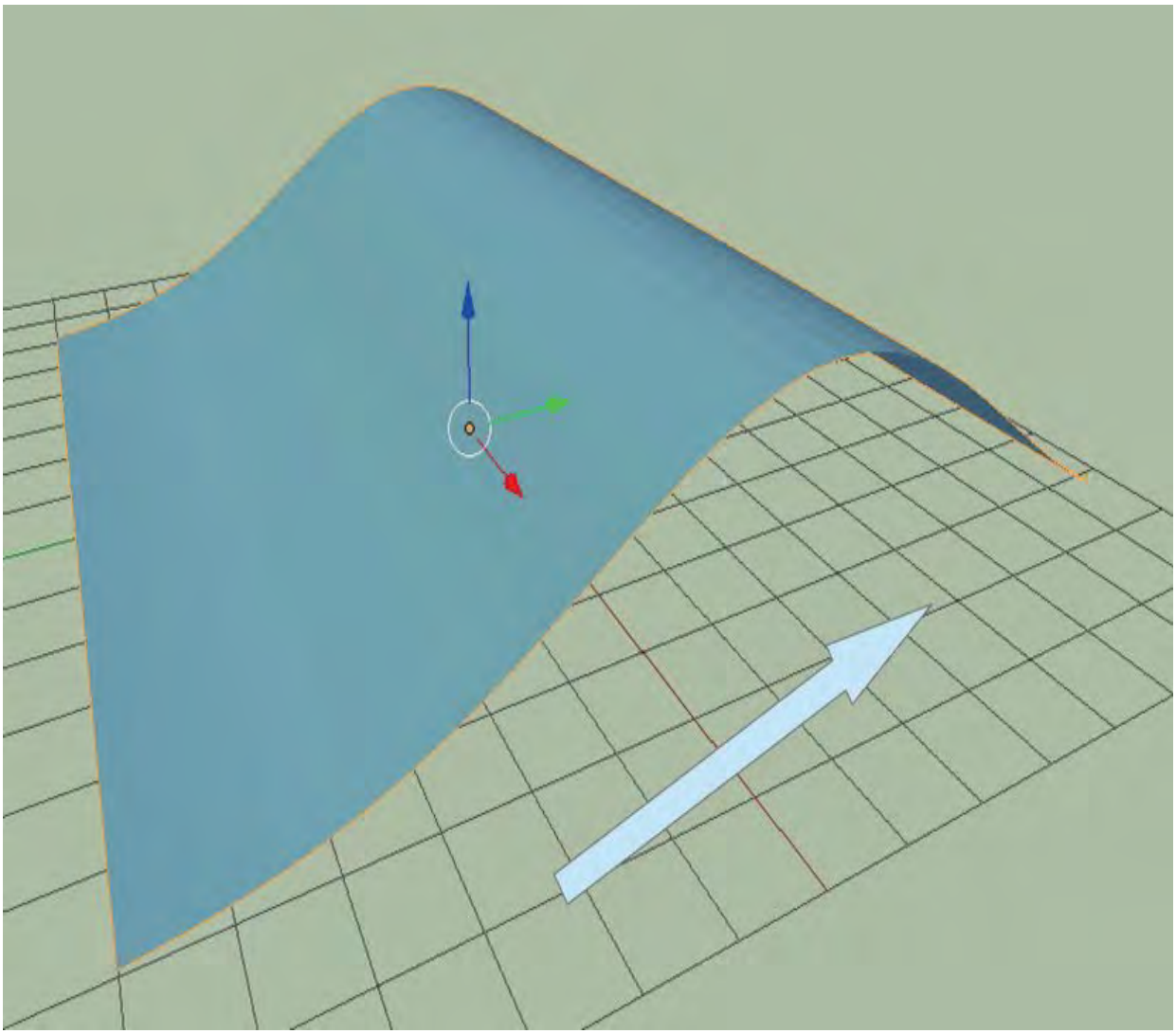
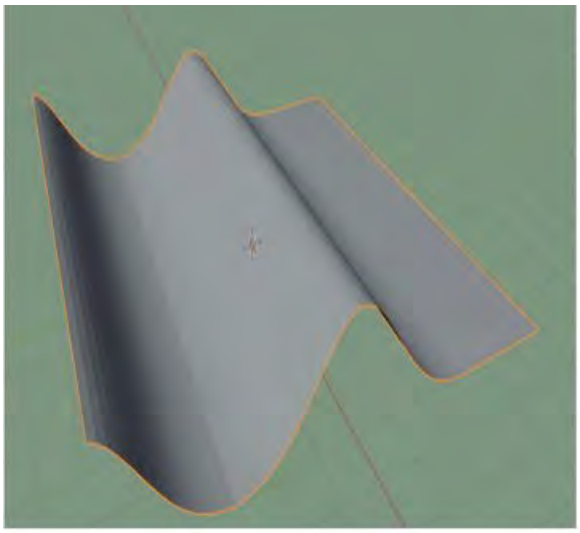


Figure 9.65

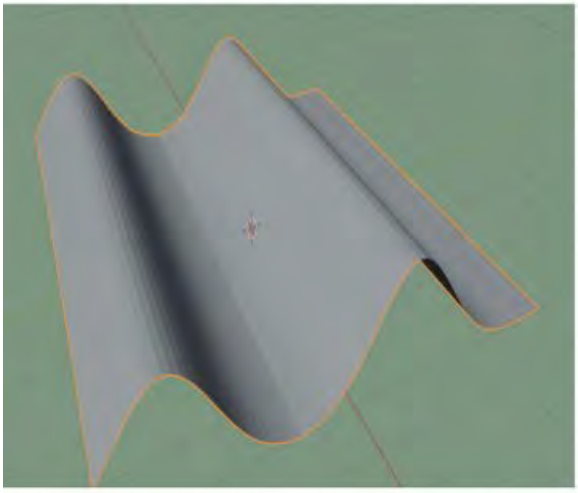
Press the **Play** button in the **Timeline Editor Header** to see the animation play.

Deselect the **X Axis** in the Modifier panel and play again. A wave along the Y axis results (Figure 9.62).

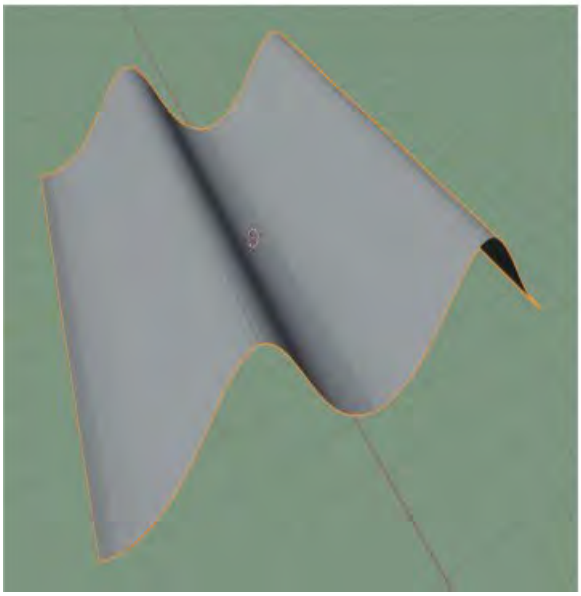
By expanding the **Start Position** and **Time** Tabs in the Modifier you have full control over the wave motion (Figure 9.63).



Frame 1



Frame 53



Frame 96

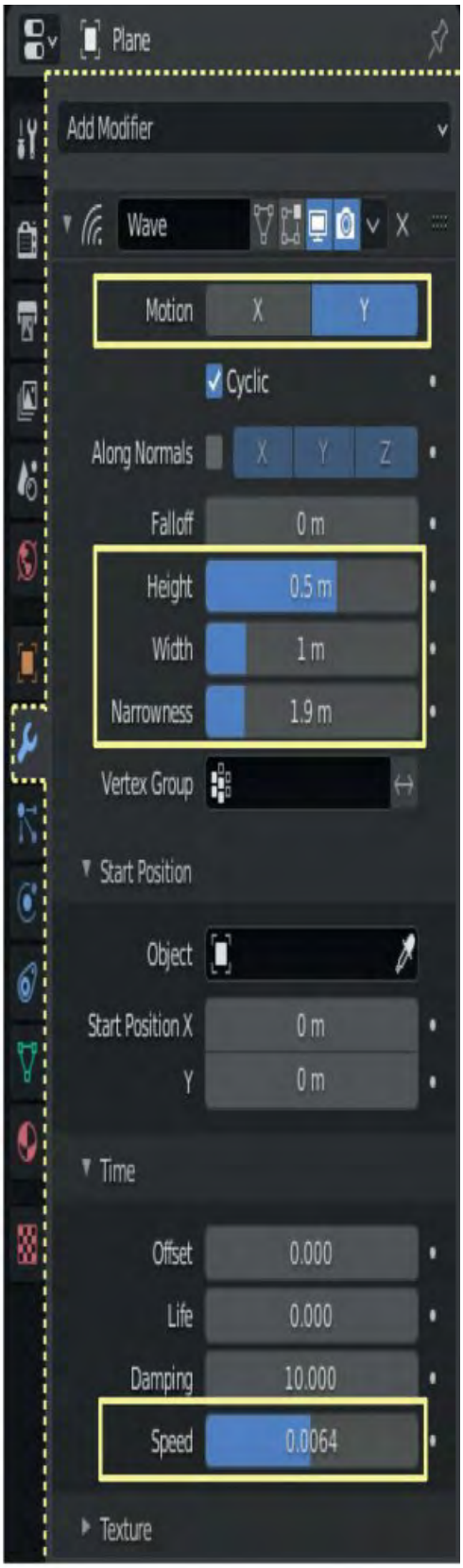


Figure 9.66

10 Editing Using Curves

10.1 Curves Circles and Paths
10.2 Bezier Curves
10.3 Bezier Circle
10.4 Nurbs Path
10.5 Nurbs Circle
10.6 Modeling from a Curve
10.7 Closed Loops
10.8 Using Nurbs Curves
10.9 Nurbs Circle
10.10 Nurbs Curve
10.11 Lofting

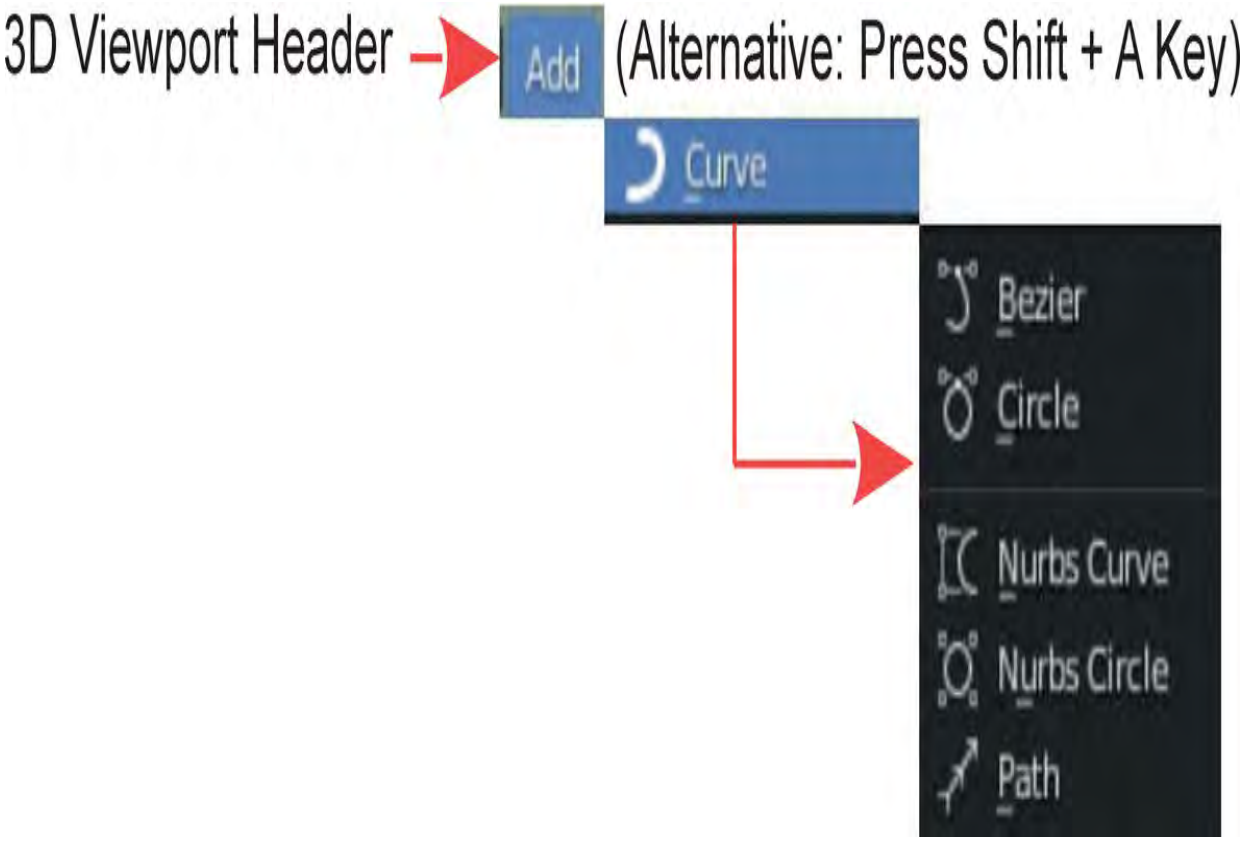


Figure 10.1

Introduction

In Blender a Curve is a line or a path used to control the shape of a mesh in modeling or the movement of an Object in animation.

In Blender there are five types of curves: **Bezier** (Bezier Curve), **Circle** (Bezier Circle), **Nurbs Curve, Nurbs Circle** and **Path.**

Each type of Curve is entered in the Blender Scene in Object Mode as a basic line. Control Handles display in Edit Mode which allow the shape of the Curve to be modified. The Curve may be Scaled and Extruded and additional control handles may be added.

A Curve does not Render. It merely acts as a control for editing a Mesh Object. The Object renders but the curve is invisible to the Render process.

Bezier and Nurbs Curves can be circular (circles) in which case they form a closed loop.

10.1 Curves Circles and Paths

In Blender, **Curves, Circles and Paths** are lines, giving a graphical representation of data or a line representing a path which controls direction or movement. Do not be confused with the Circle Object.

Curves are editable, which means that the shape of the curve may be altered to suit a particular application. An Object can be made to follow a Curve in an animation or it can be extruded along a Curve to affect its shape or it can be duplicated along a Curve. **Curve Circles** are merely circular Curves joined at the ends forming a continuous loop.

In Blender there are five basic Curve options which are accessed in the **Add menu in the 3D Viewport Editor Header** (Figure 10.2) or by pressing the **Shift + A Key** with the Mouse Cursor in the 3D Viewport Editor.

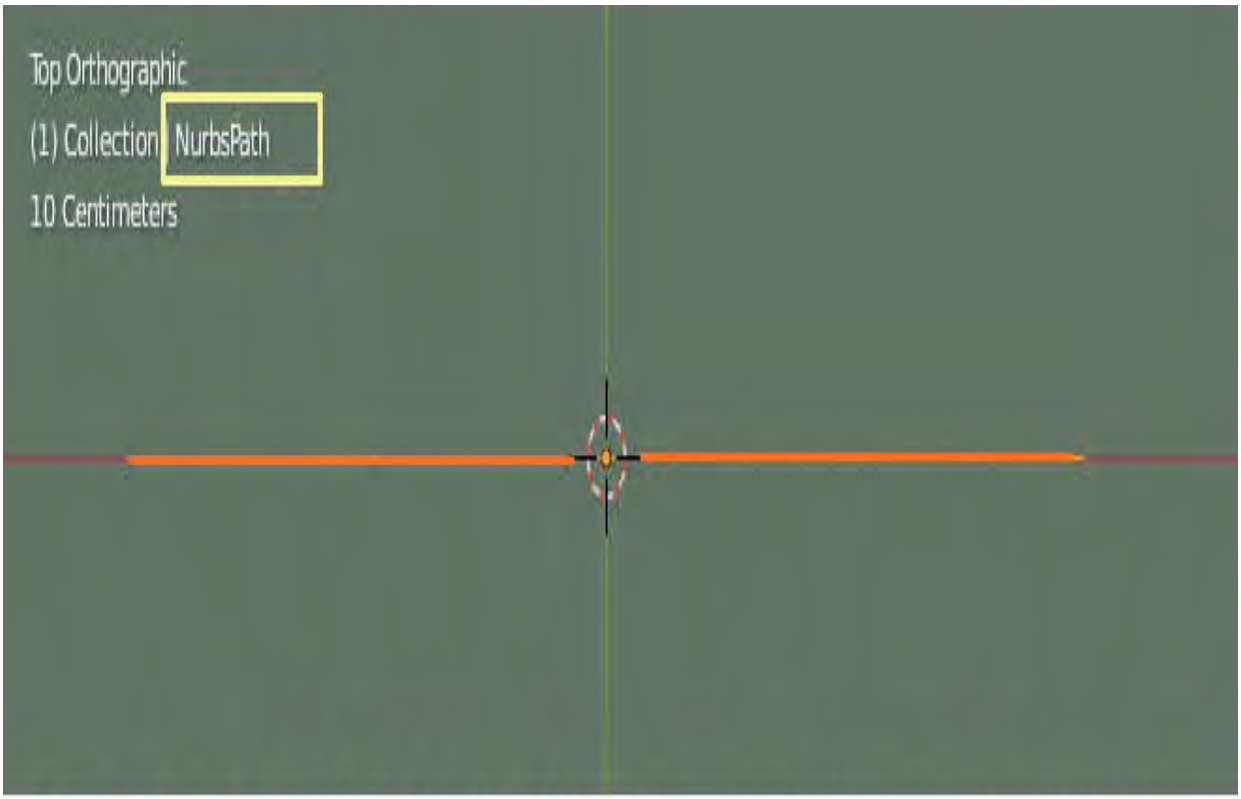
To examine the options place the 3D Viewport Editor in Top Orthographic View and delete the default Cube.

Add
Object
Mesh
Curve
Surface
Metaball
Text
Armature
Lattice
Empty
Grease Pencil
Speaker
Camera
Light
Light Probe
Force Field
Collection Instance
Bezier
Circle
Nurbs Curve
Nurbs Circle
Path
Draw Curve

Figure 10.2

Path: To understand Curves, Circles and Paths begin with the **Path** option. Select **Path** from the **Add — Curve** menu.

A **Nurbs Path** is entered in the 3D Viewport Editor in **Object Mode (Top Orthographic View).**



Edit Mode: In Edit Mode the Path displays yellow with **Control Point**

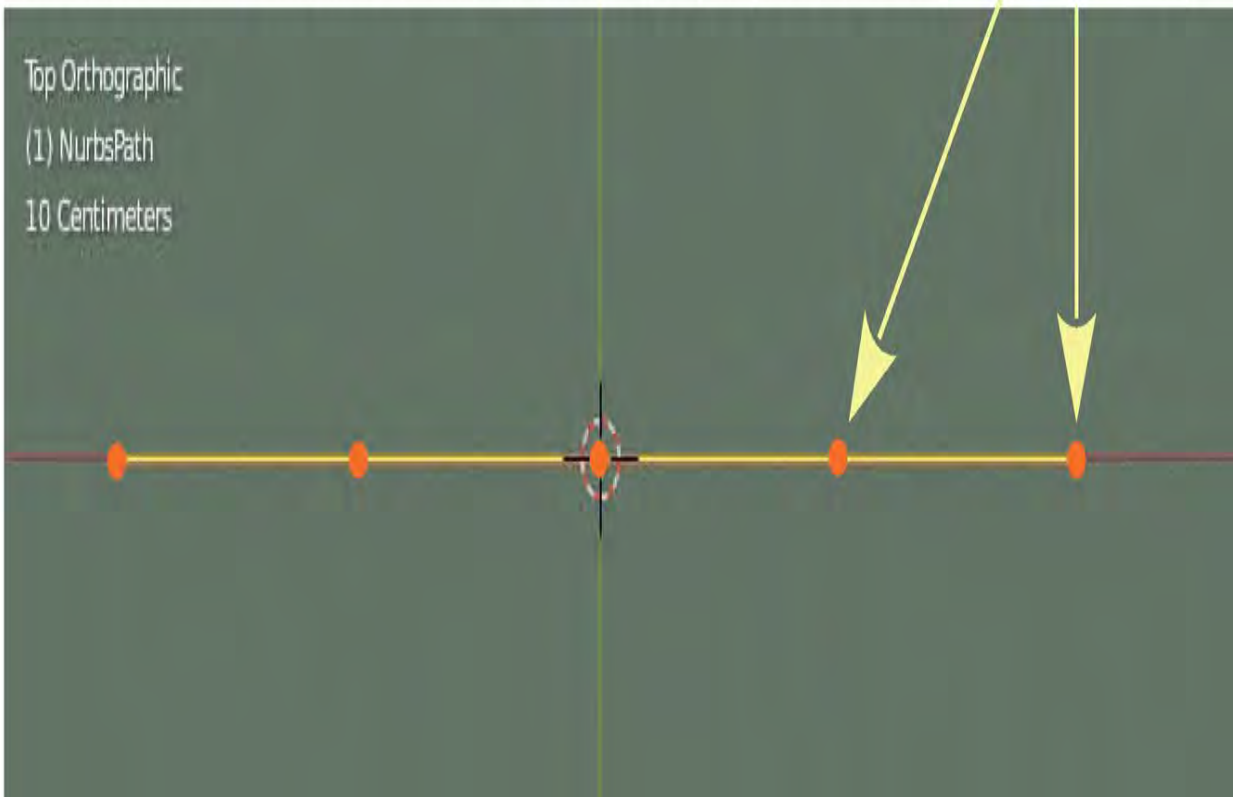
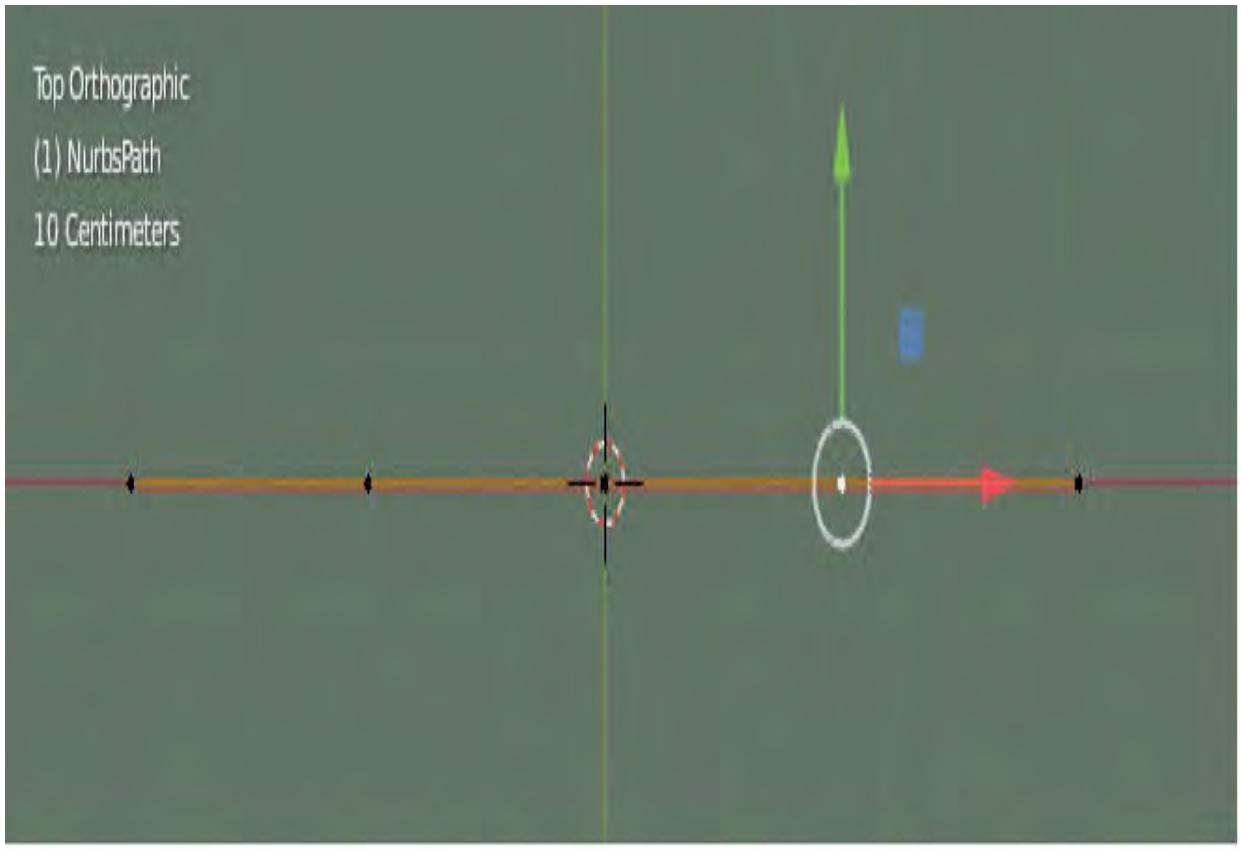


Figure 10.3

In Edit Mode, deselect the Path and click LMB on a **Control Point** (turns white).

Press G Key (Grab), move the Mouse or activate the Translate Tool in the Tool Panel to move the Control Point reshaping the Path.


Top Orthographic
(1) NurbsPath
10 Centimeters

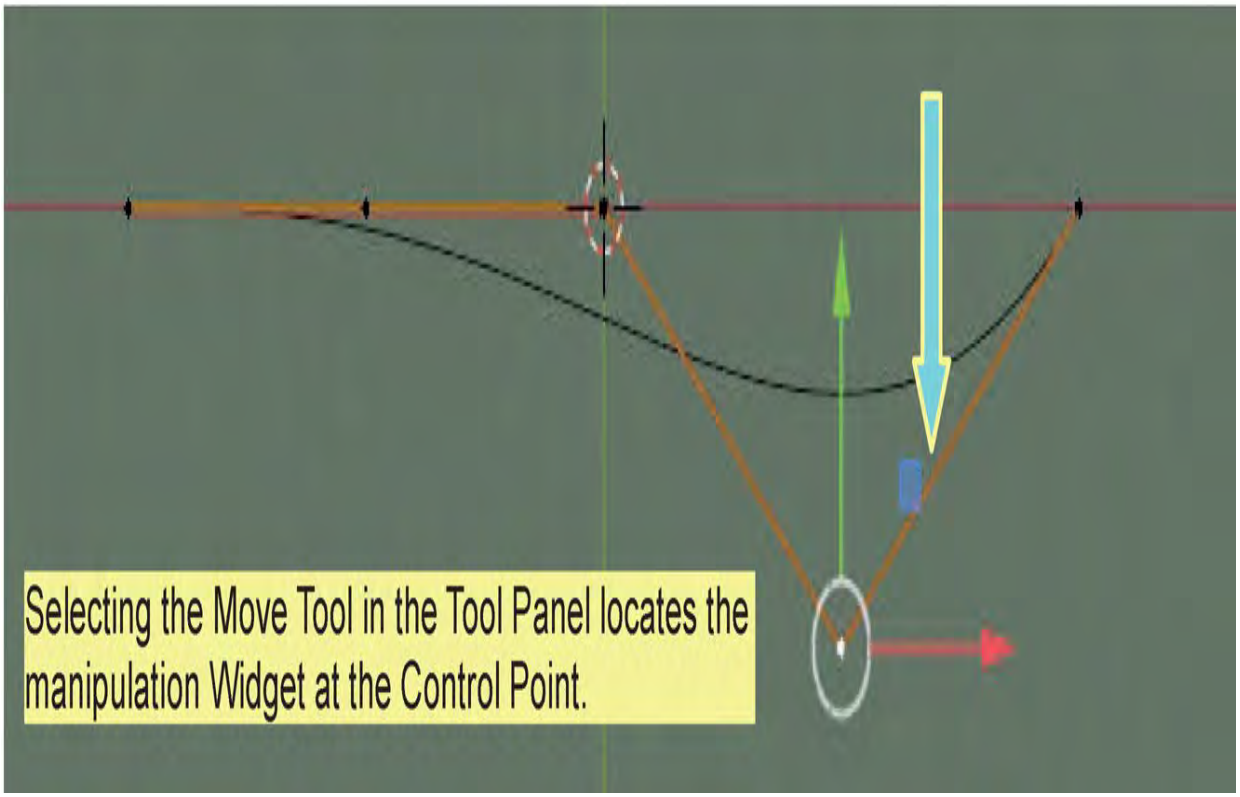

Selecting the Move Tool in the Tool Panel locates the manipulation Widget at the Control Point.

Figure 10.4

One application for a **Path,** is to animate an Object to move, following the Path. In this case, checking **Normals** in the **Edit Mode, Viewport Overlays,** displays chevrons along the Path

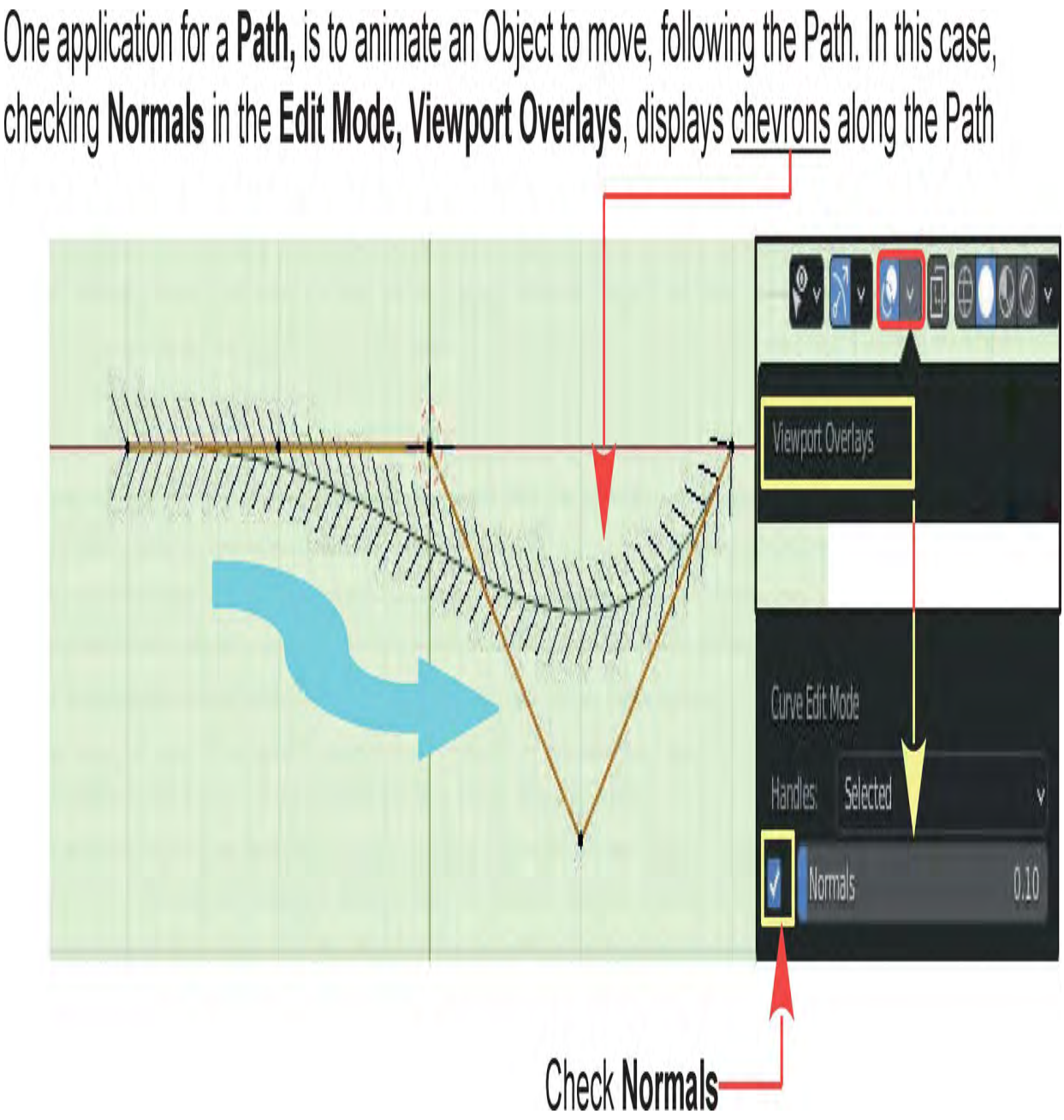


Figure 10.5

The **chevrons** indicate the direction of travel when the Object is Animated to follow the Path.

Remember: The Path may be Scaled, Rotated and Translated like any Object but it does NOT Render when an Image or Animation is created.

How to Animate an Object to follow a Path is described in Chapter 20 — 20.28.

10.2 Bezier Curve

In some respects the Bezier Curve appears to be similar to a Curve Path when reshaped as shown in the previous example (Figure 10.5). Inspection in Edit Mode reveals a more refined control for shaping the curve.

Chevrons display on the Curve in Edit Mode with Normals checked in Overlays.

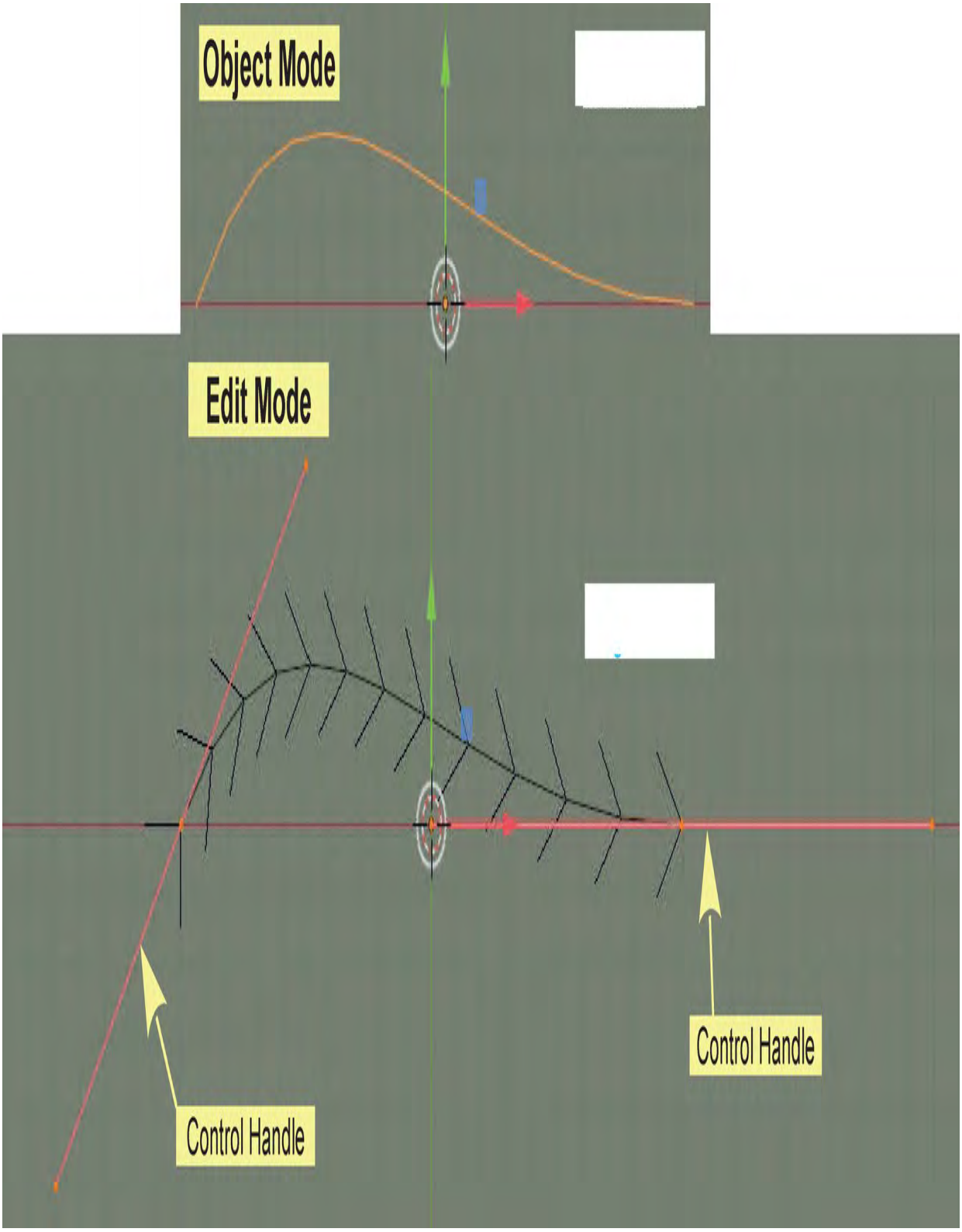
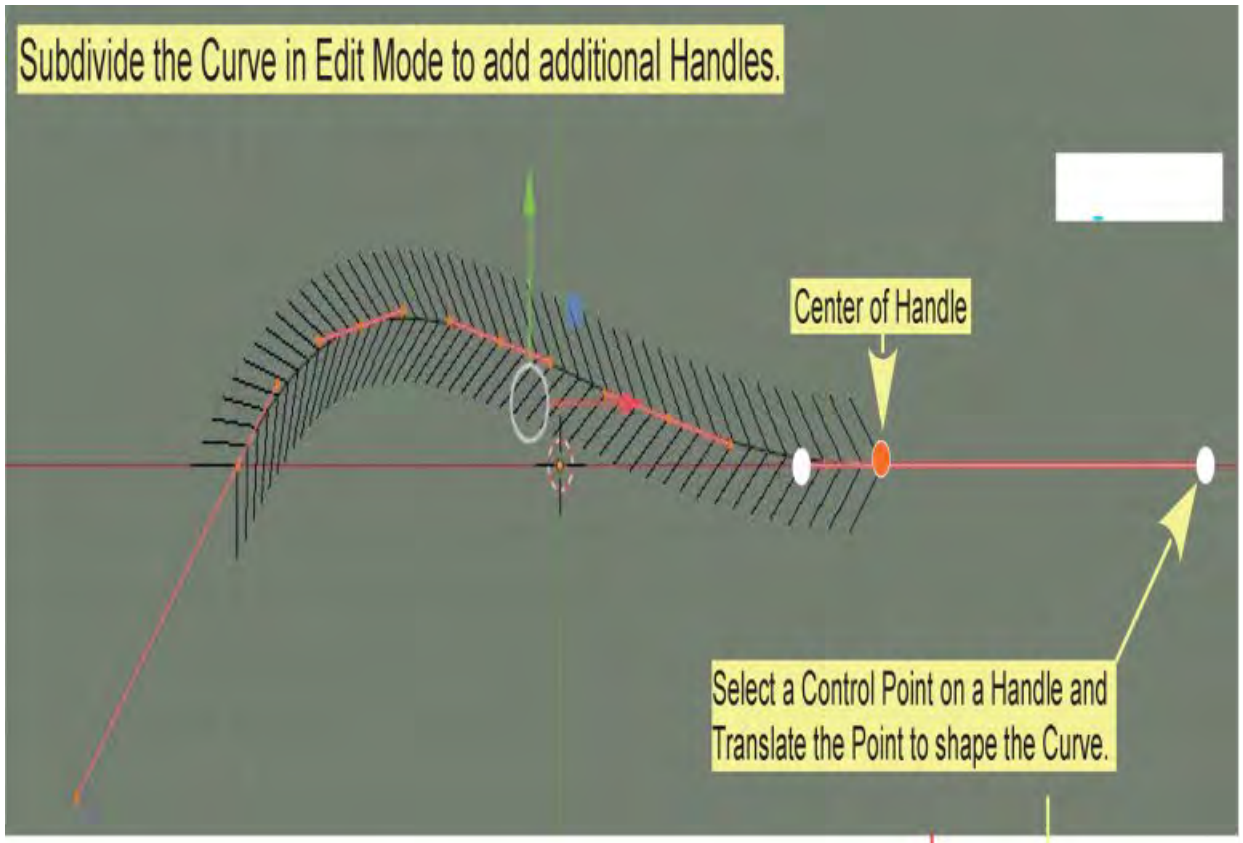


Figure 10.6

Figure 10.7


Subdivide the Curve in Edit Mode to add additional Handles.
Center of Handle
Select a Control Point on a Handle and
Translate the Point to shape the Curve.

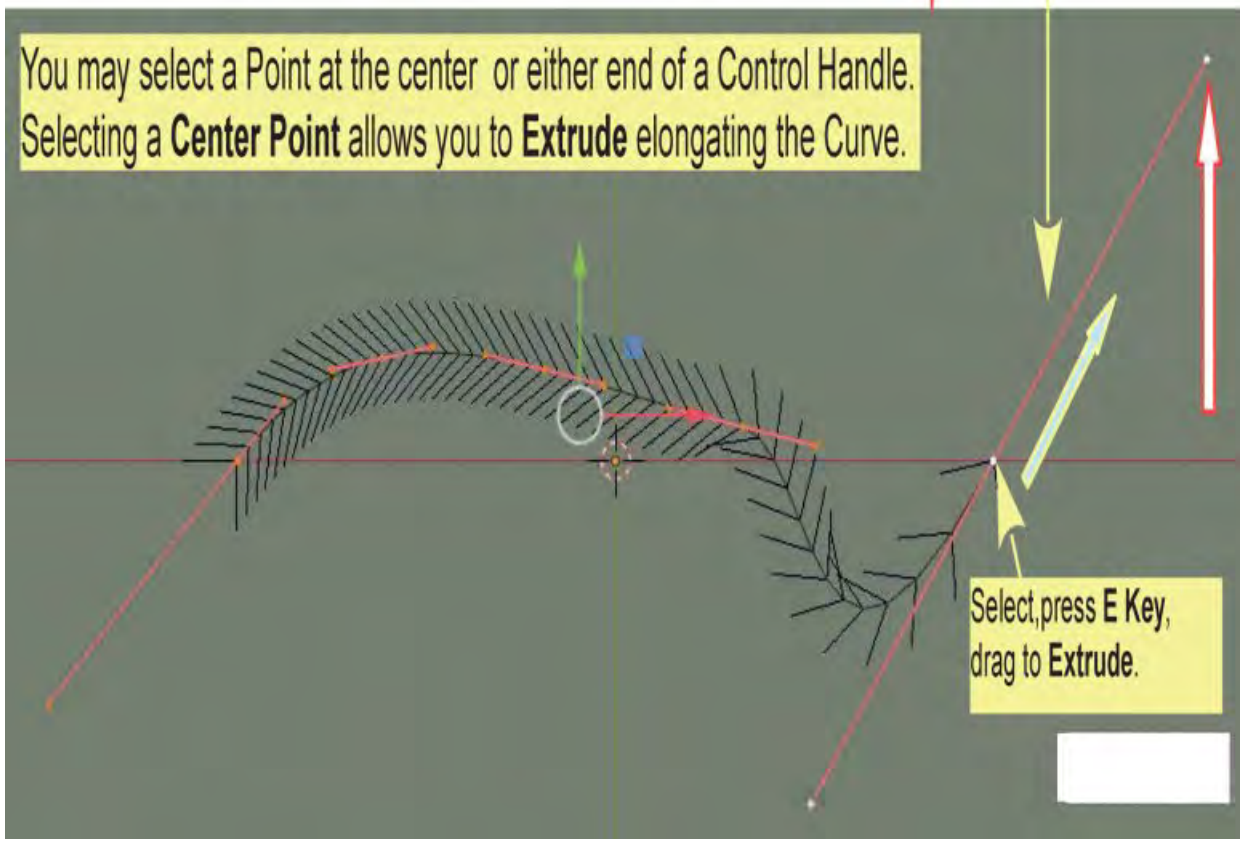

You may select a Point at the center or either end of a Control Handle.
Selecting a **Center Point** allows you to **Extrude** elongating the Curve.
Select,press **E Key**,
drag to **Extrude**.

Figure 10.8
Figure 10.9

10.3 Bezier Circle

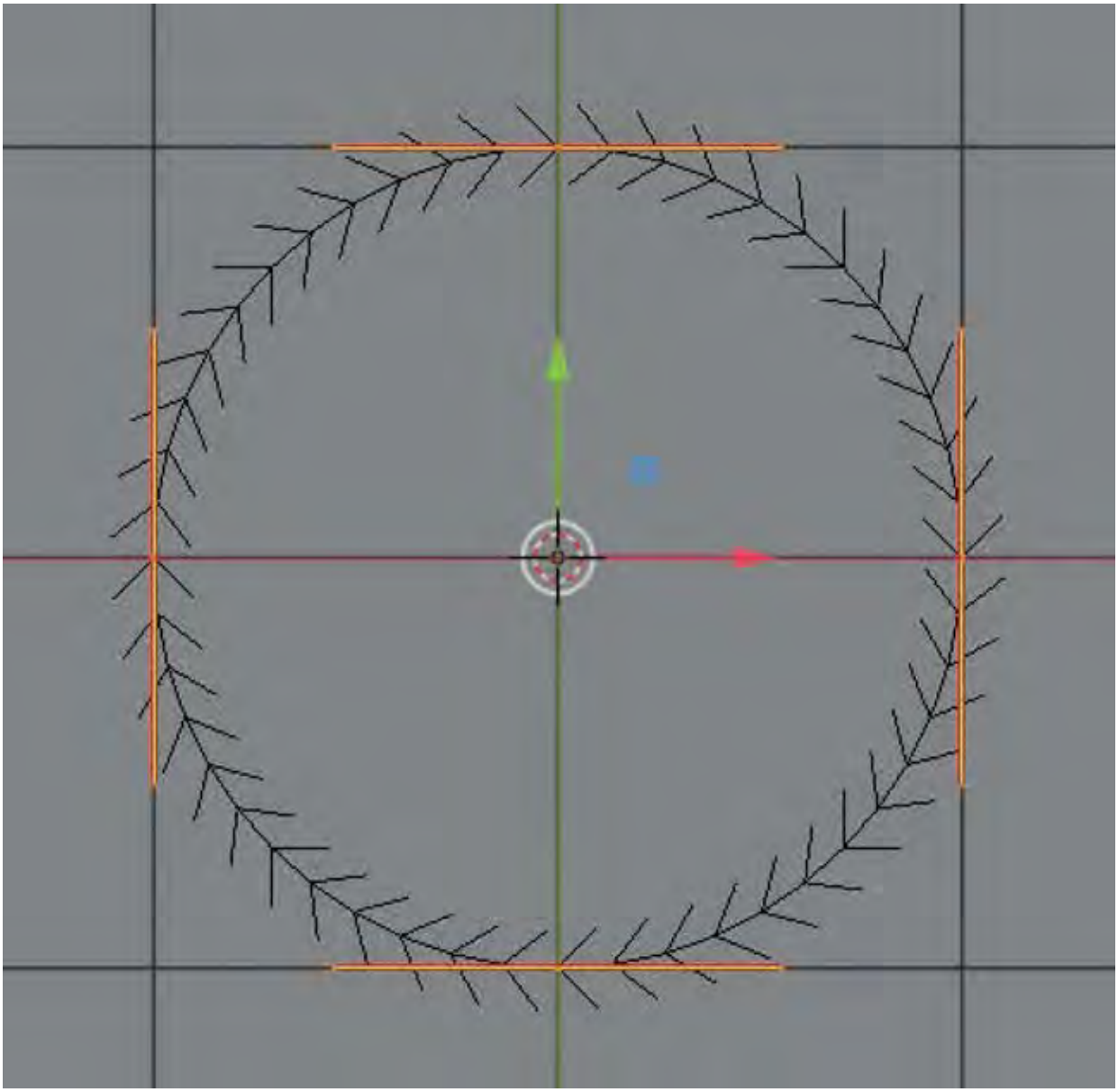


Figure 10.10

The **Bezier Circle** is similar to the Bezier Curve with control handles at the four cardinal points.

Click RMB to select a handle. **G Key** to grab and move to reshape the path. **R Key** to rotate and flatten the curve.

Note: The Bezier Circle, Nurbs Path and Nurbs Circle are shown in Edit Mode.

10.4 Nurbs Path

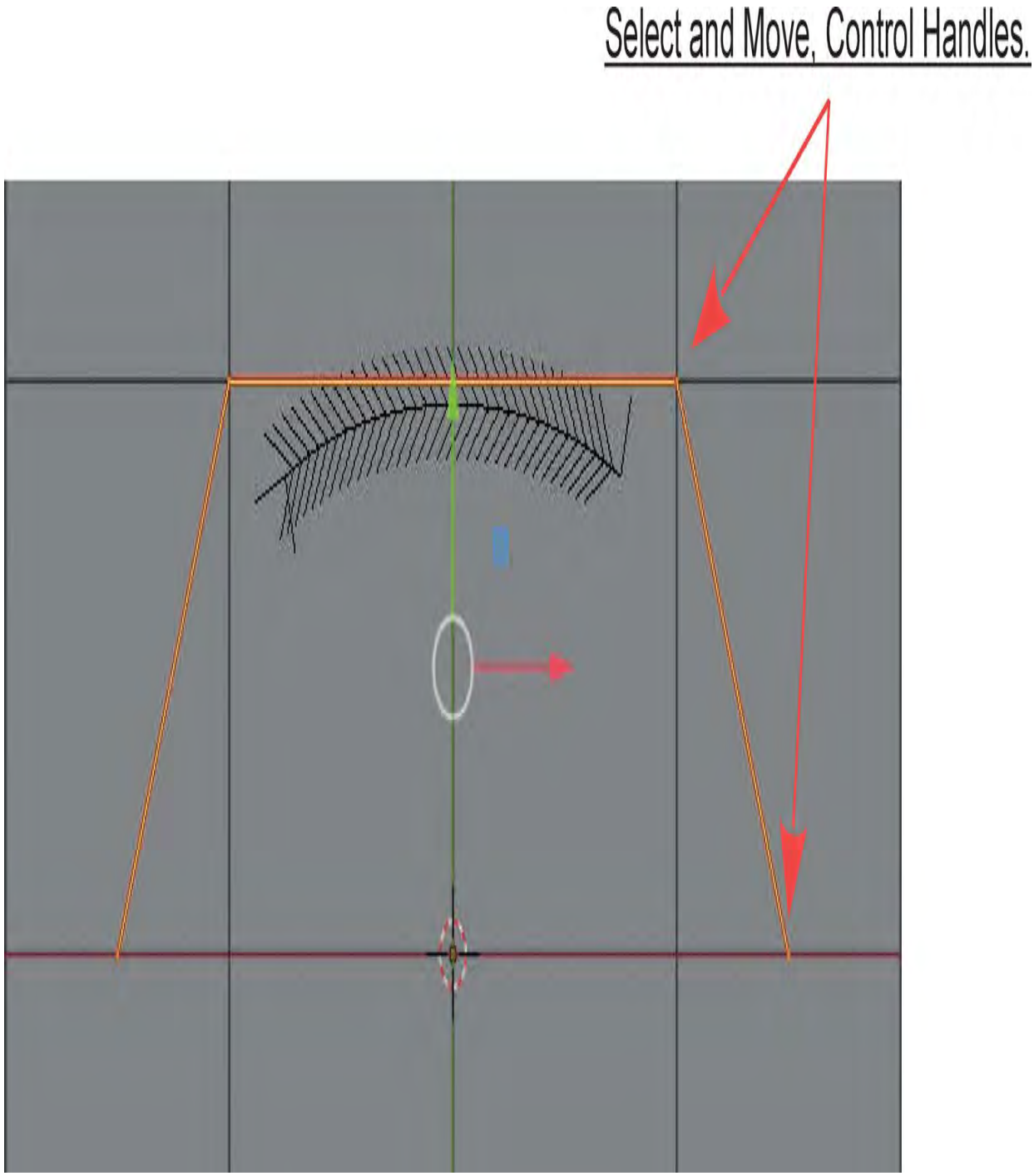


Figure 10.11

10.5 Nurbs Circle

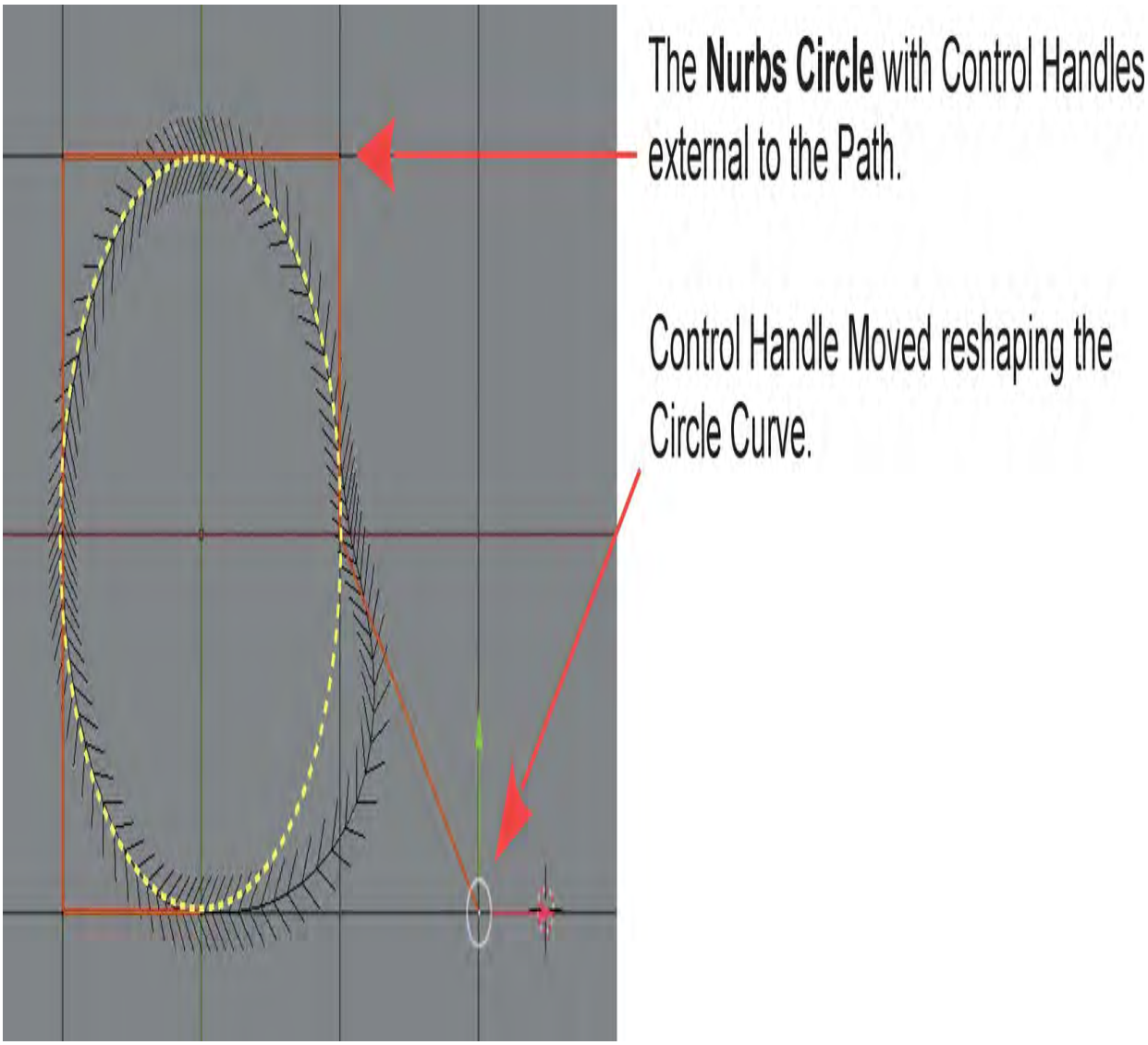


Figure 10.12

Note: The diagrams serve only to show you what the different Curves look like and how to Edit their shape.

10.6 Modeling from a Curve

Any Curve Path or Circle may be used to create a Mesh Object by Extrusion.

Enter a **Bezier Curve** in **User Perspective View,** zoom in and **Tab into Edit Mode.**

Note: The following is **NOT** the way to create a Mesh Object but to demonstrate what happens when the Curve is Extruded in Edit Mode. When Edit Mode is entered the Control Handles at both ends of the Curve are selected. Extrude down on the Z Axis (E Key+Z Key, drag the Mouse).

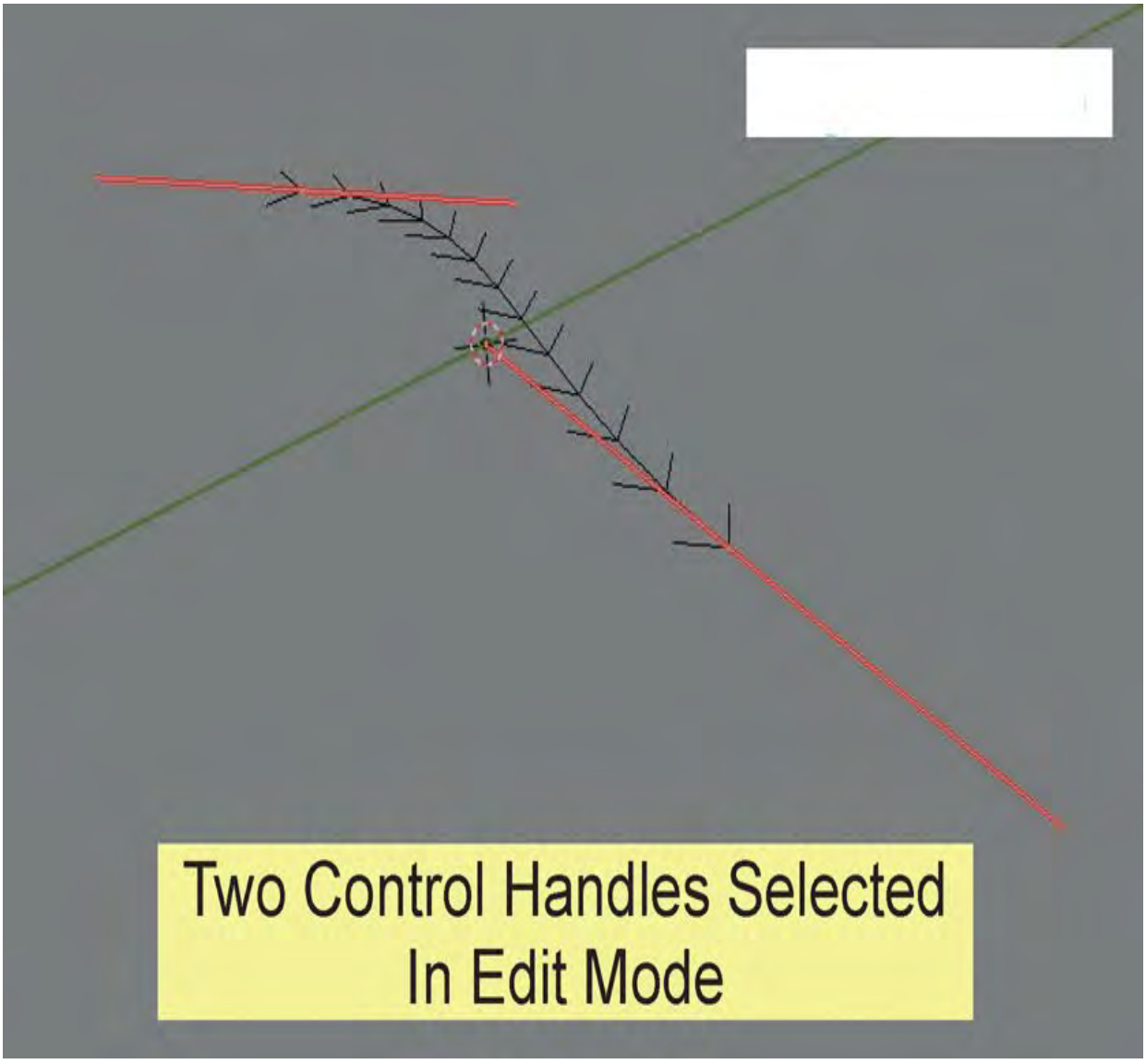


Figure 10.13

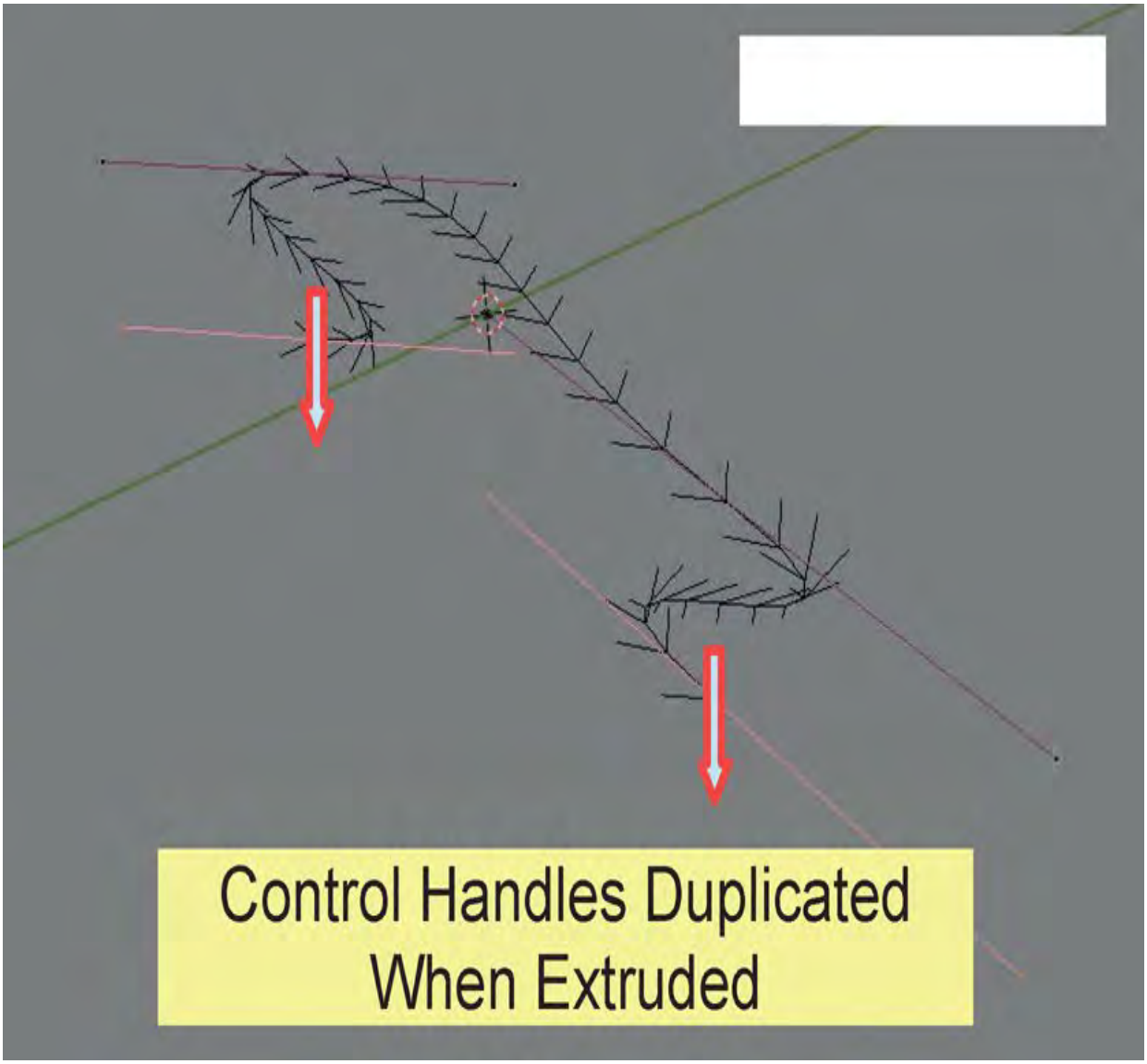


Figure 10.14

To create a Mesh from a Curve be in **Object Mode** with the default **Bezier Curve** selected **(NOT Extruded).** In the **Properties Editor, Object Data Properties, Geometry Tab** change values as shown (Figure 10.15). Rotate the 3D Editor Viewport (Figure 10.16)

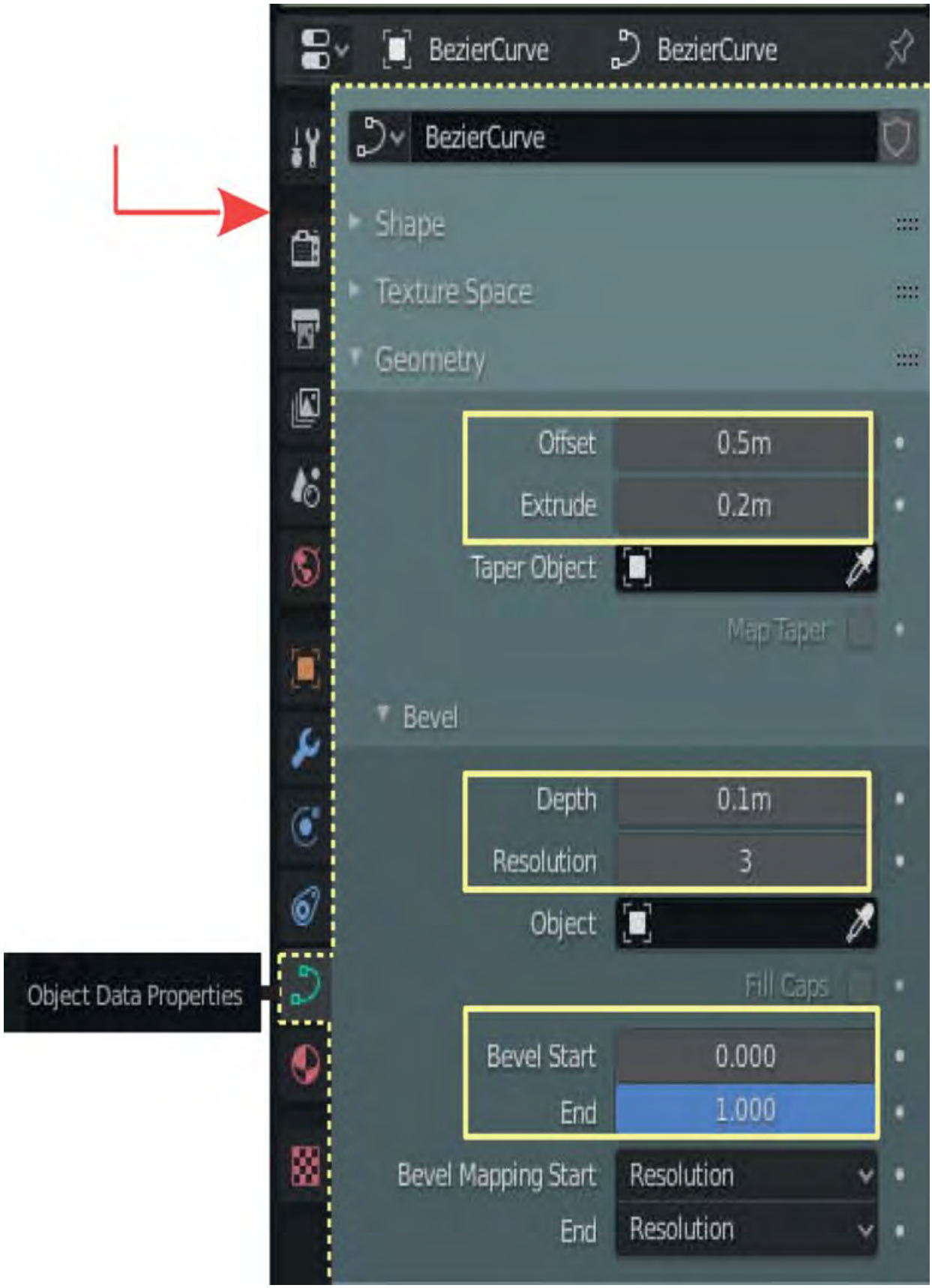

BezierCurve
BezierCurve
BezierCurve
Shape
Texture Space
Geometry
Offset
0.5m
Extrude
0.2m
Taper Object
Map Taper
Bevel
Depth
0.1m
Resolution
3
Object
Fill Caps
Bevel Start
0.000
End
1.000
Bevel Mapping Start
Resolution
End
Resolution
Object Data Properties

Figure 10.15

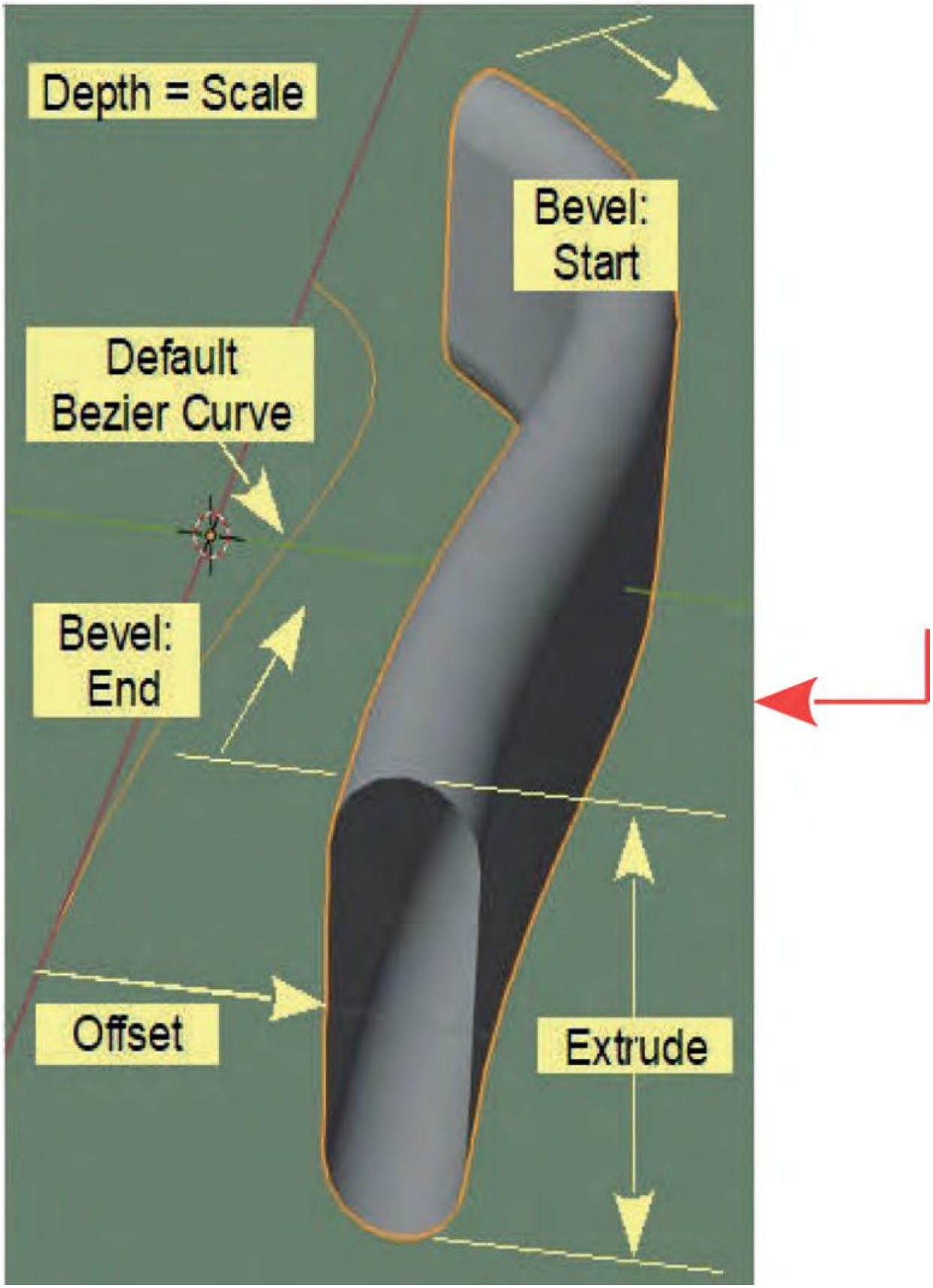

Depth = Scale
Bevel:
Start
Default
Bezier Curve
Bevel:
End
Offset
Extrude

Figure 10.16

With the shape of the Mesh created, in Object Mode, go to the **3D Viewport Editor Header,** click on **Object** and select **Convert - Mesh.** In Edit Mode you will see that Vertices, Edges and Faces have been created.

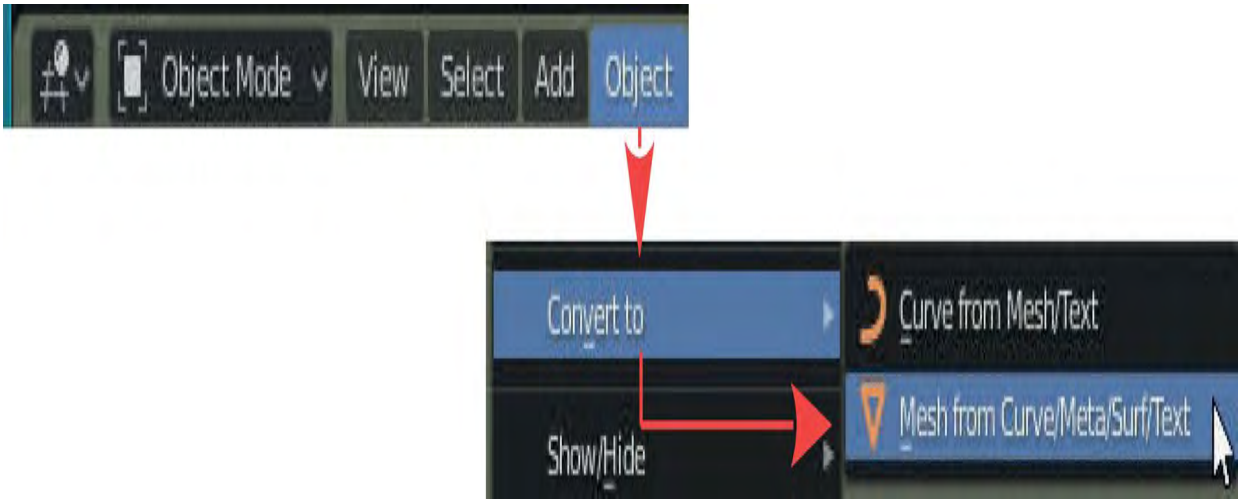


Figure 10.17 3D Viewport Editor Header

Note: Before converting to a Mesh, in Edit Mode, you can select the Control Handles on the original Bezier Curve and reshape the Object. Once converted to a Mesh the original Curve is deleted.

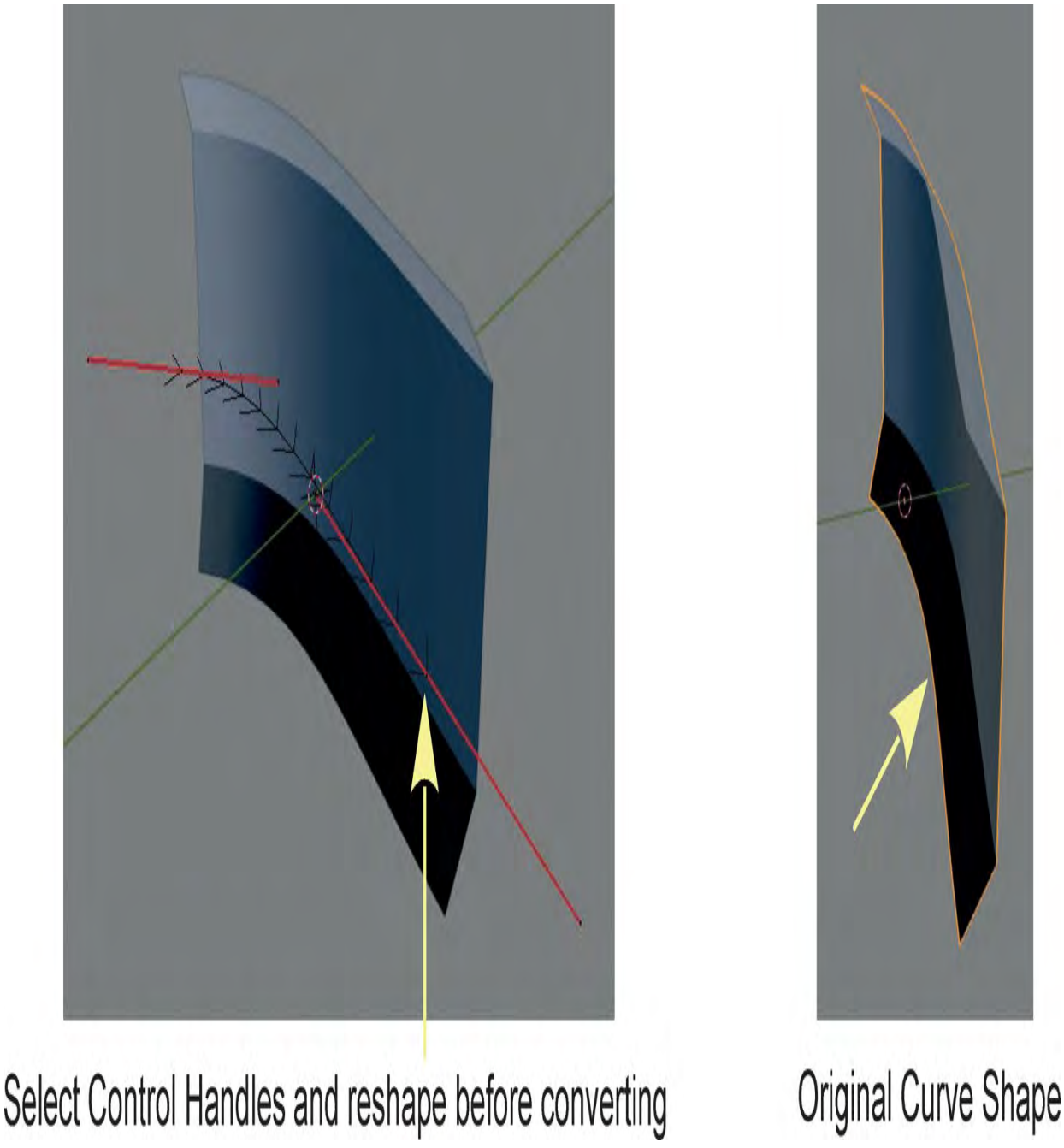


Figure 10.18

10.7 Closed Loops

Bezier and Nurbs Circles are Closed Loops which means they can be used to create tubular Objects or form a continuous path for an animation. Any Curve or Path may be converted to a Closed Loop

The following shows a **Curve Path** entered in **Edit Mode** with the RH Control Point Moved then Extruded three times. **In Edit Mode,** click **Curve** in the Header and select **Toggle_Cylic** to form a closed loop.

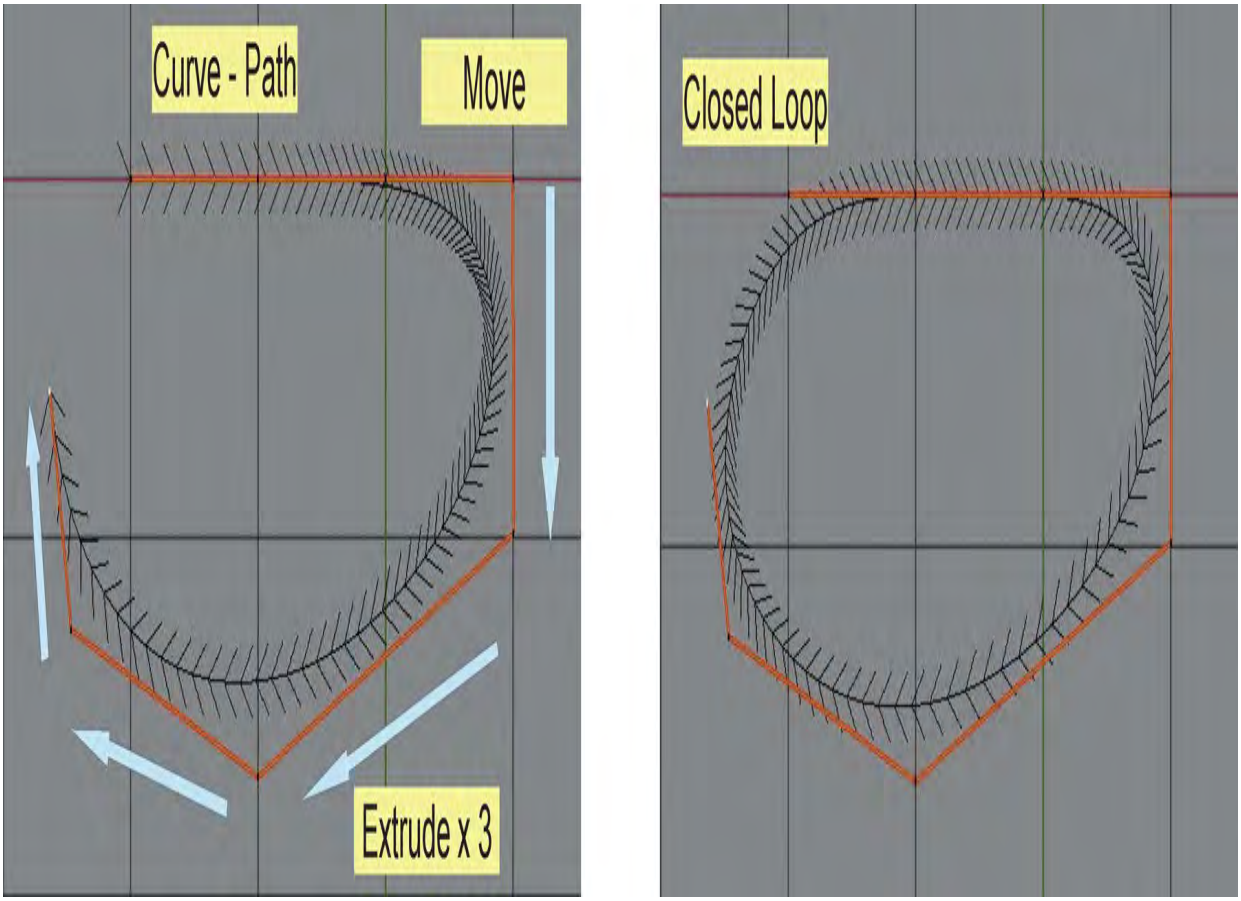


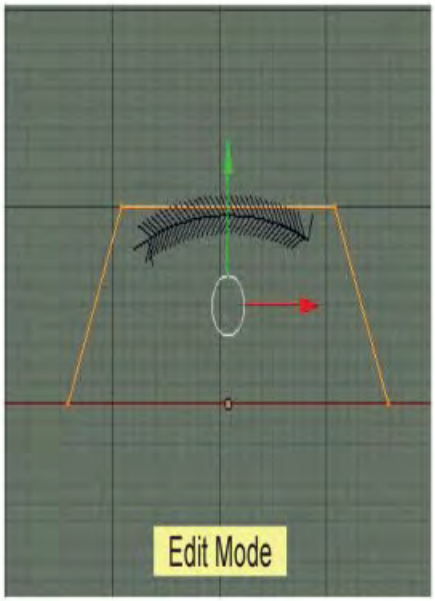
Figure 10.19

10.8 Using Nurbs Curves

Nurbs Curves are entered in the 3D Viewport Editor by pressing **Shift + A Key** or clicking **Add** in the 3D Viewport Header and selecting from the menu that displays. There are two options; **Nurbs Curve** and **Nurbs Circle** (Figure 10.20). **Place the 3D Viewport Editor in Top Orthographic View.**

Menu

Nurbs Curve



Nurbs Circle

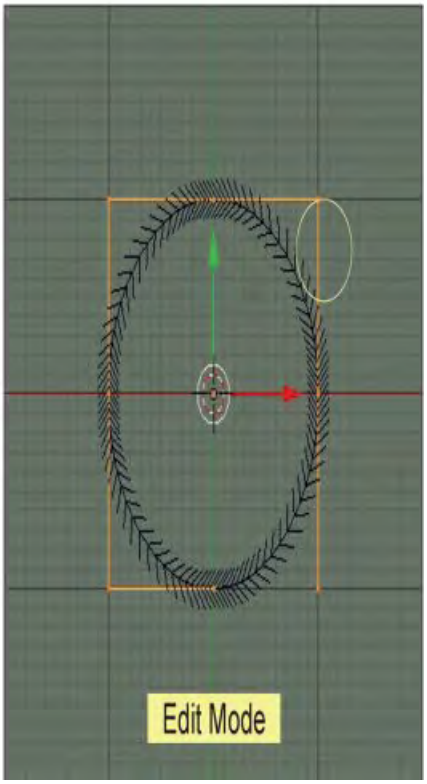


Figure 10.20

In Edit Mode, Nurbs Curves and Circles have control handles attached with which you manipulate the shape (Figure 10.21). In both cases you see chevrons spaced on the Curve indicating that they may be used as Paths for animation.

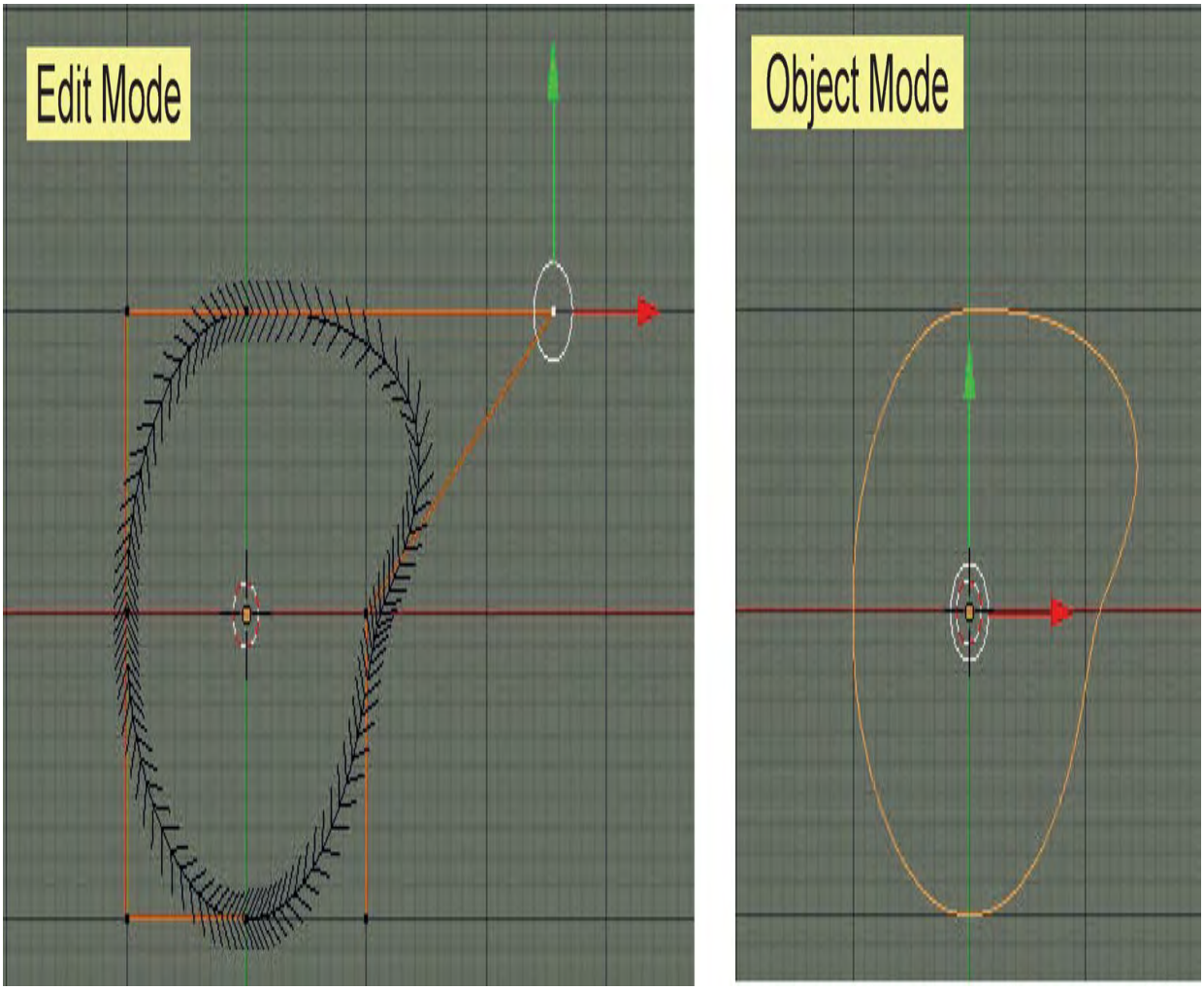


Figure 10.21

10.9 Nurbs Circle

Figure 10.22 shows a **Nurbs Circle** with a Control Point selected and Moved on the X Axis.

With the Nurbs Circle selected in **Object Mode** the shape may be expanded by changing settings in the **Properties Editor, Object Data Properties, Geometry Tab** (Figure 10.22). To demonstrate, skew the 3D Viewport into a User Orthographic View as shown in Figure 10.22

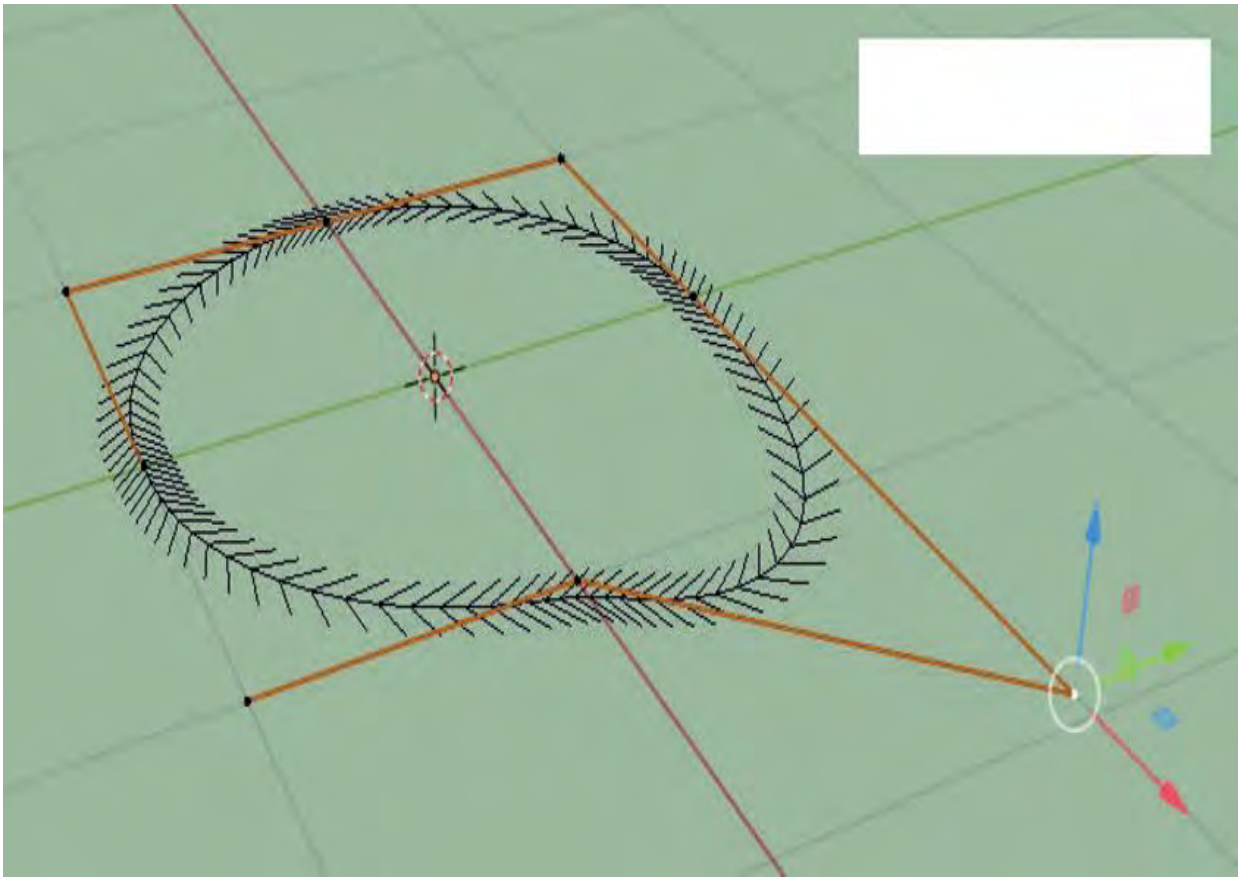
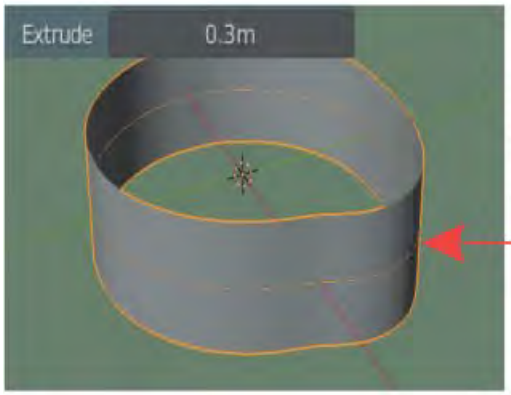
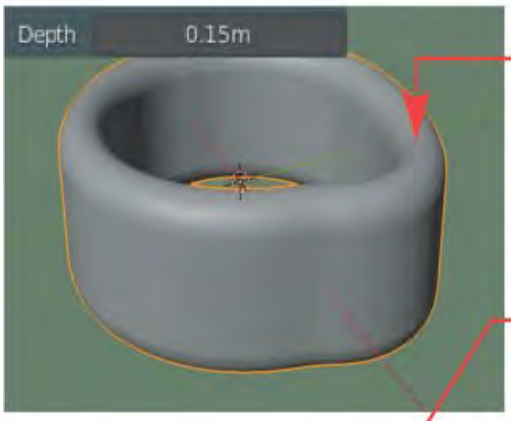


Figure 10.22



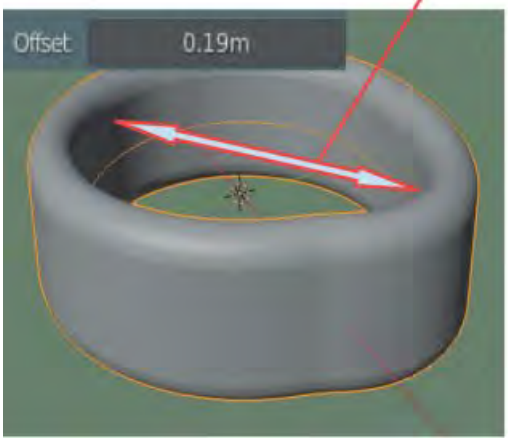
Extrude extends the Curve Circle into a shape.



Bevel – Depth bevels the edge.

In Edit Mode the Control Points remain in place.

Geometry – Offset increases the size on the X and Y axis.



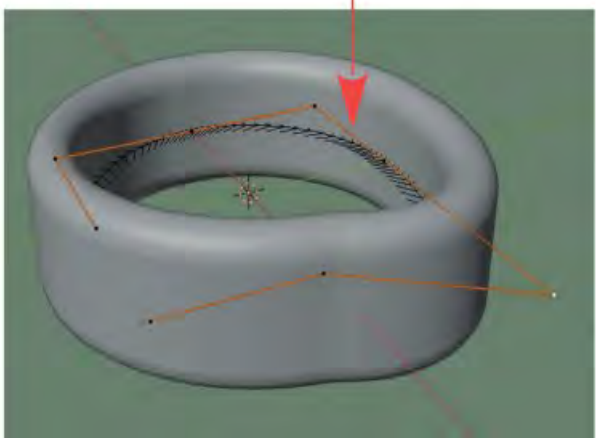
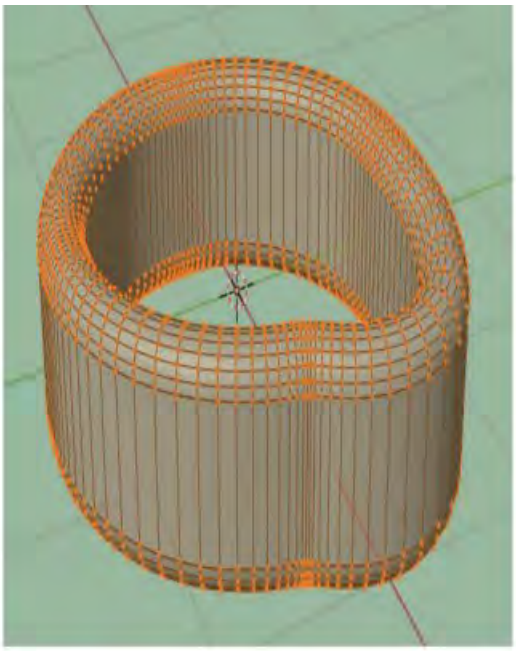


Figure 10.23

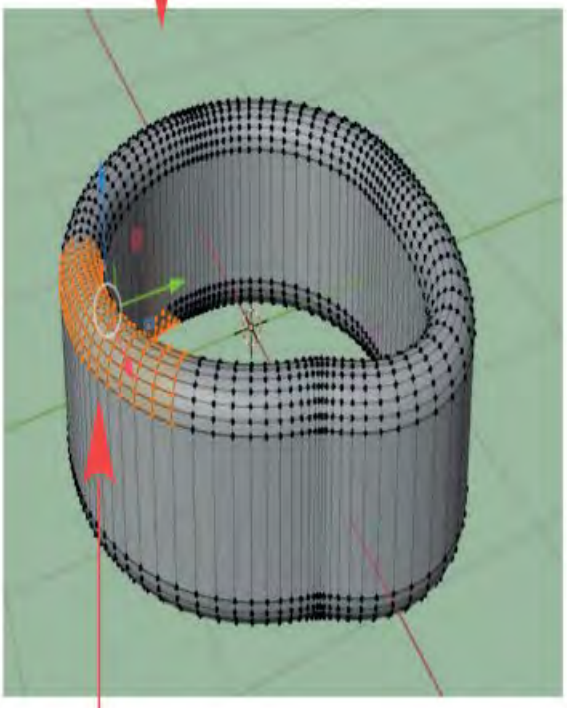
With the Circle expanded then selected in **Object Mode** you may convert the shape into a **Mesh Object (** in the Header, click Object, select Convert to — Mesh from Curve). You see the Object in Edit mode with vertices, edges and faces (Figure 10.24).

Note: When you convert to a Mesh Object the ability to use the settings in the Properties Editor, Data buttons, Geometry tab is no longer available.
Deselect the Vertices then select individual vertices for manipulation.

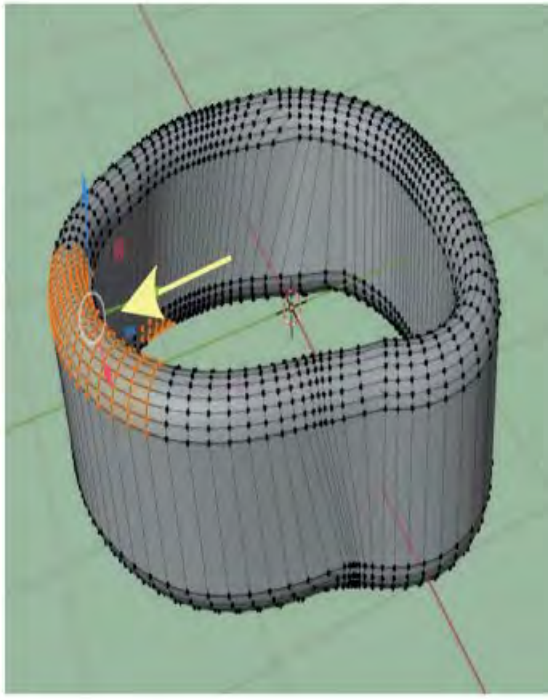
Deselect the Vertices then select individual vertices for manipulation.



A key - Deselect



Vertices Selected

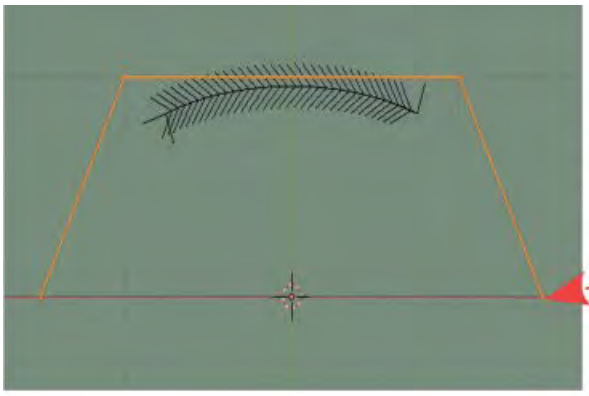


Vertices Translated – Proportional Editing Enabled

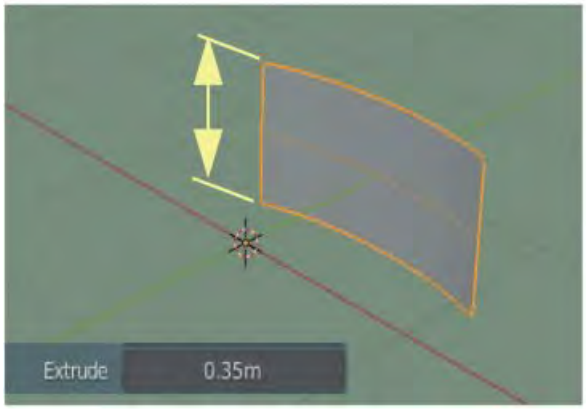
Figure 10.24

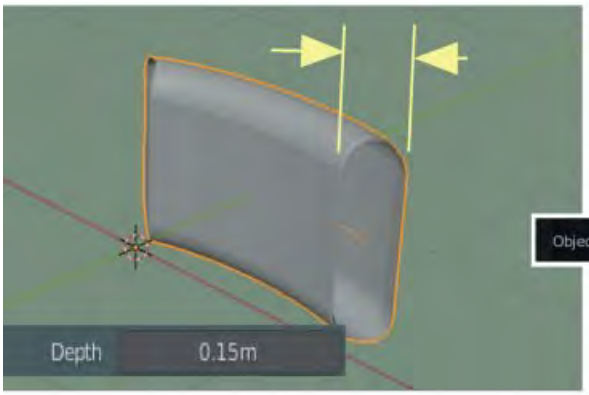
This procedure shows that by converting one type of object to another, you have different options for shape manipulation.

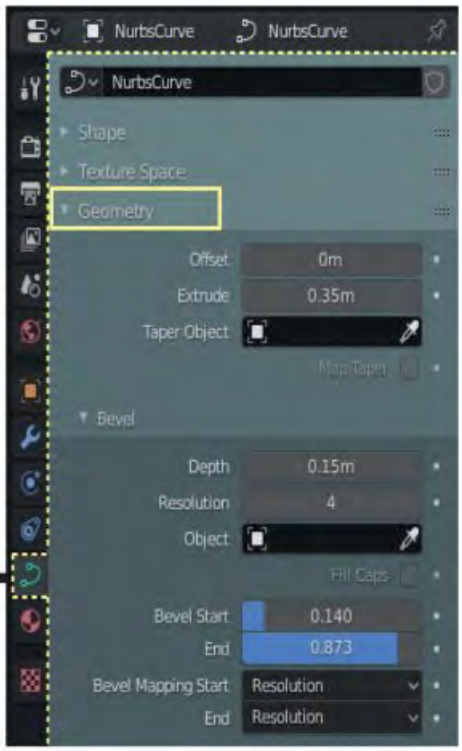
10.10 Nurbs Curve



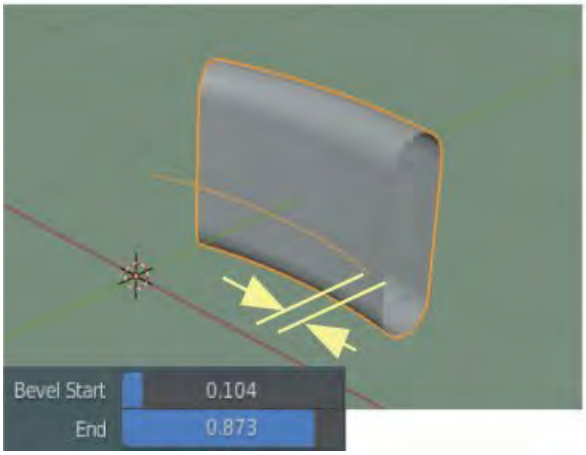
Edit mode shows the Curve surrounded by Control Handles (orange lines) with control points (orange dots)







Object Data Properties



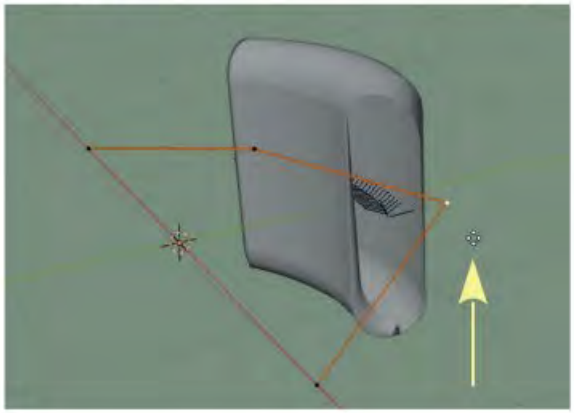


Figure 10.25

The **Nurbs Curve** is another starting place for creating shapes, objects or animation paths.

You may select a single point or multiple points then Move, Rotate or Scale to shape the Curve.

In **Object Mode** settings in the **Properties Editor, Object Data Properties, Geometry Tab** control the shape.

Move Control Handles to alter the shape in **Edit Mode.**

10.11 Lofting

Lofting is sometimes referred to as **Lathing** which is the process of generating shapes using Curves. The shape is generated then converted to a Mesh Object.

To demonstrate the process a **Bezier Circle** will be used in conjunction with a **Bezier Curve.**

Begin a new Blender Scene, delete the default Cube then add a Bezier Circle. Deselect the circle and add a Bezier Curve. Zoom in on the 3D Viewport Editor (Figure 10.26).

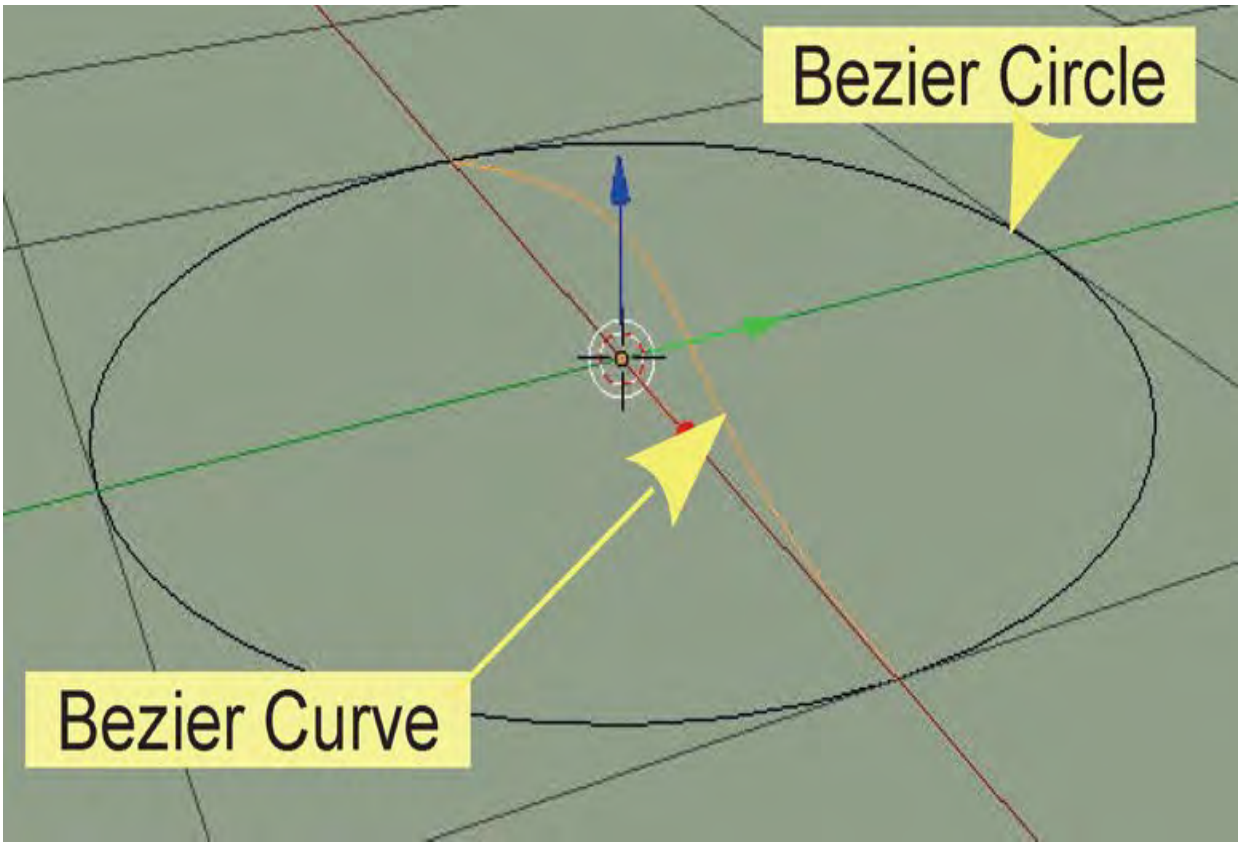


Figure 10.26

Deselect the Curve and select the Circle. In the **Properties Editor, Object Data Properties, Geometry Bevel Tab,** click on **Taper Object** and select **Bezier Curve** (Figure 10.27). Check **Fill Caps.**

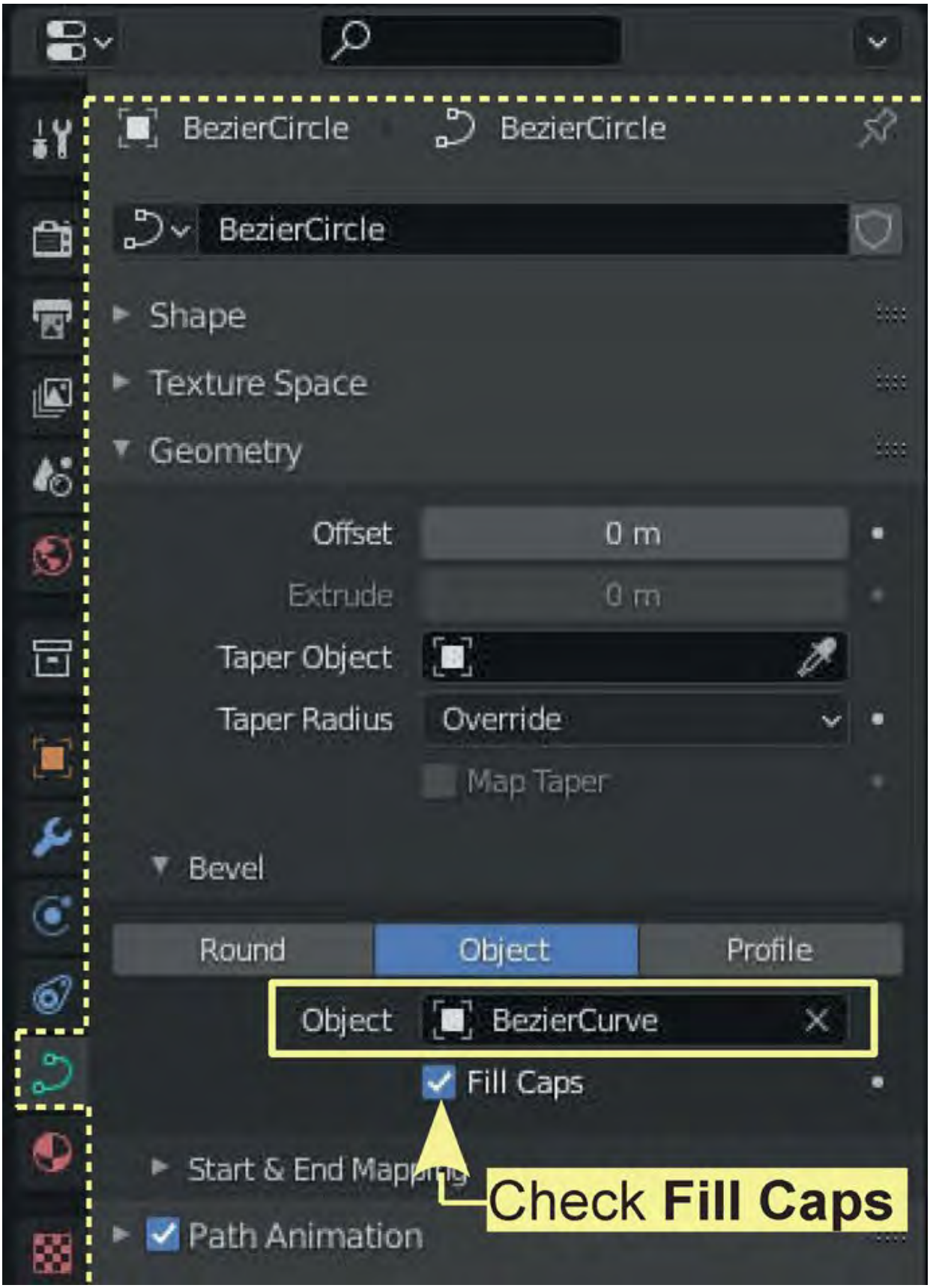

BezierCircle
BezierCircle
BezierCircle
Shape
Texture Space
Geometry
Offset
0 m
Extrude
0 m
Taper Object
Taper Radius
Override
Map Taper
Bevel
Round
Object
Profile
Object
BezierCurve
Fill Caps
Start & End Mapping
Path Animation
Check Fill Caps

Figure 10.27

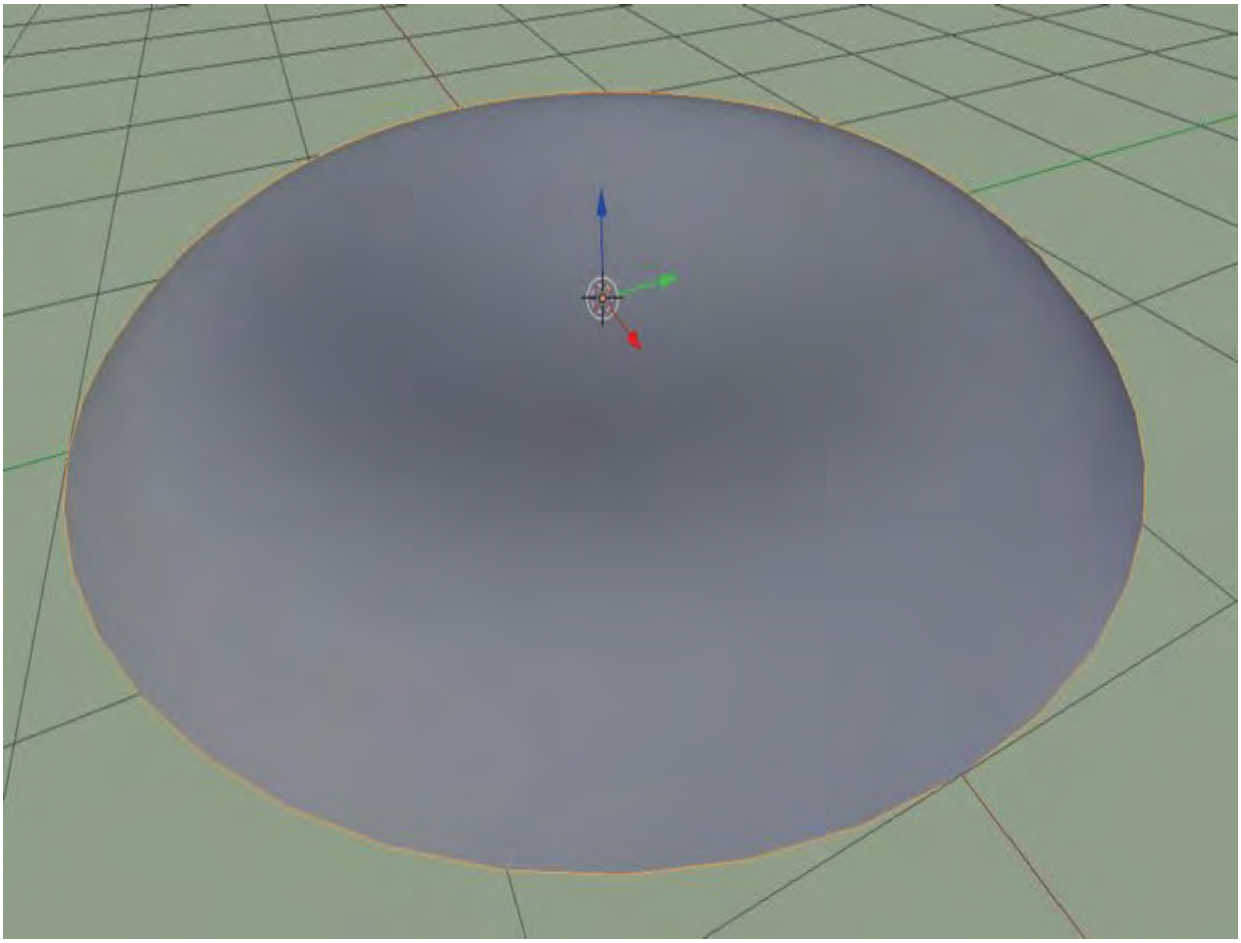


Figure 10.28

A shape is generated in the 3D Viewport Editor (Figure 10.28).

To understand what has occurred place the 3D Viewport Editor in **Wireframe Viewport Shading** Mode (Figure 10.29). Select the **Bezier Curve** by clicking on the name in the **Outliner Editor** (Figure 10.30). To see more clearly turn off the grid and floor display in the **Overlays.**

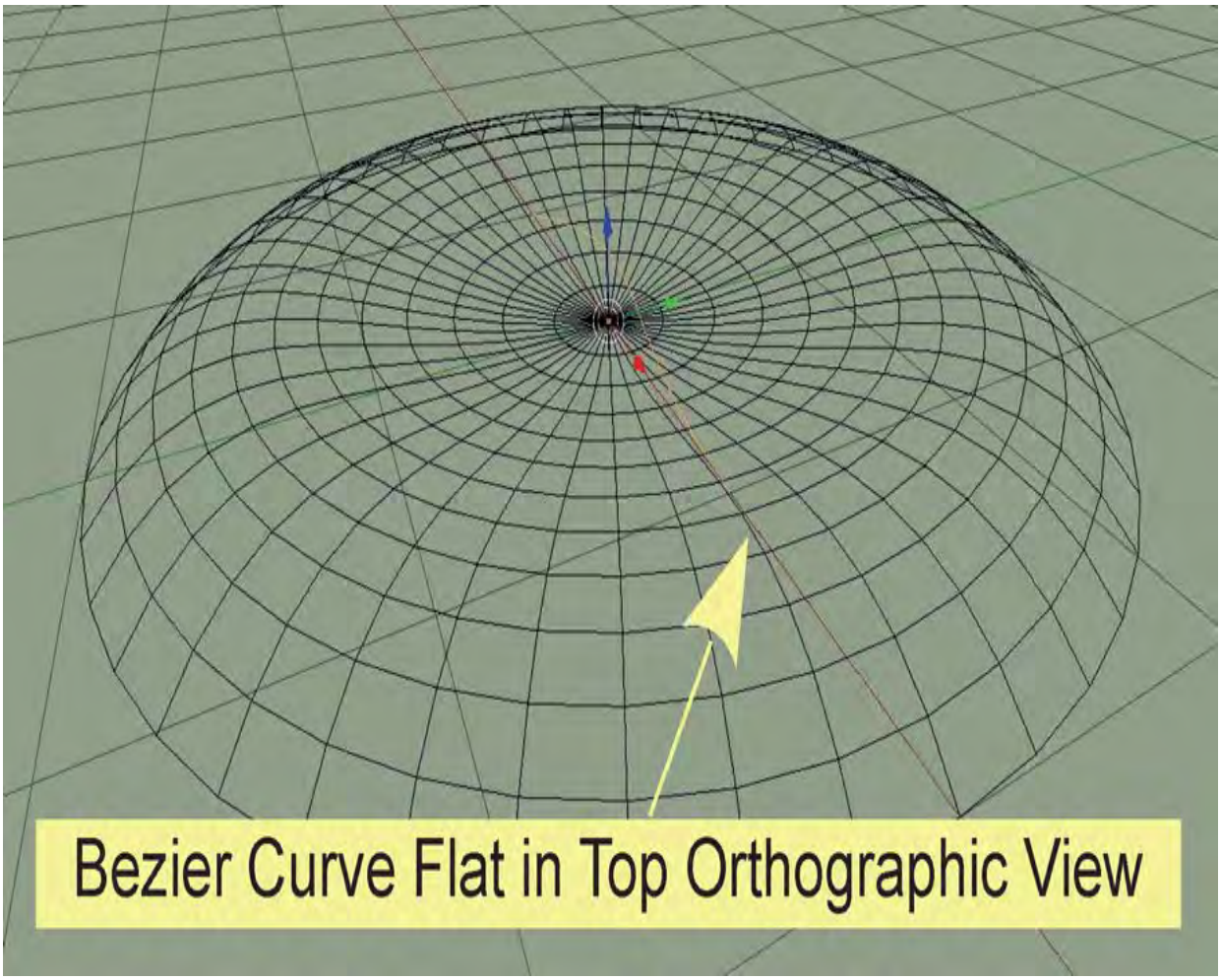


Figure 10.29

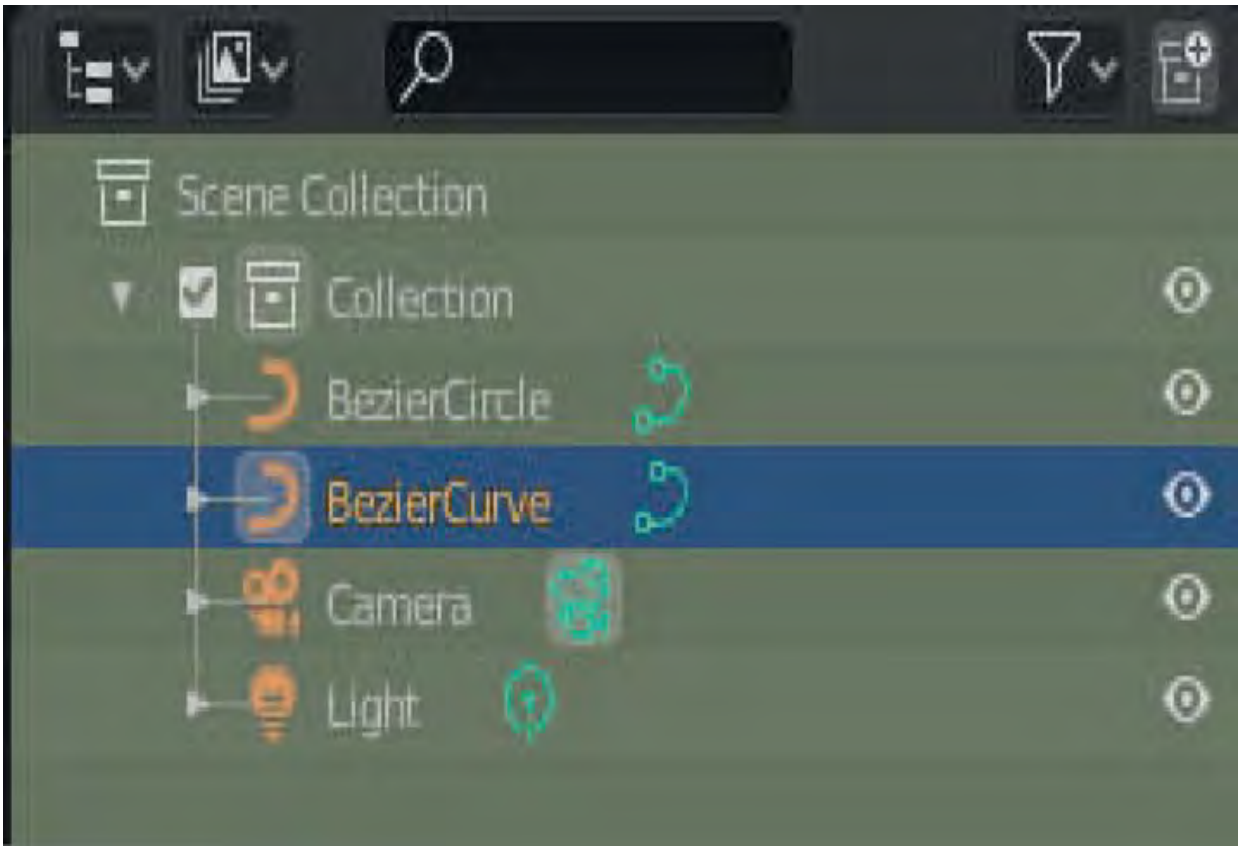


Figure 10.30

The profile of the Bezier Curve is presented flat in **Top Orthographic View** (Figure 10.26). VVith the curve selected press R + X + 90 + Enter to flip it on edge. Go into **Front Orthographic View** and translate the Curve to align with the profile of the generated shape (Figures 10.31 & 10.32.

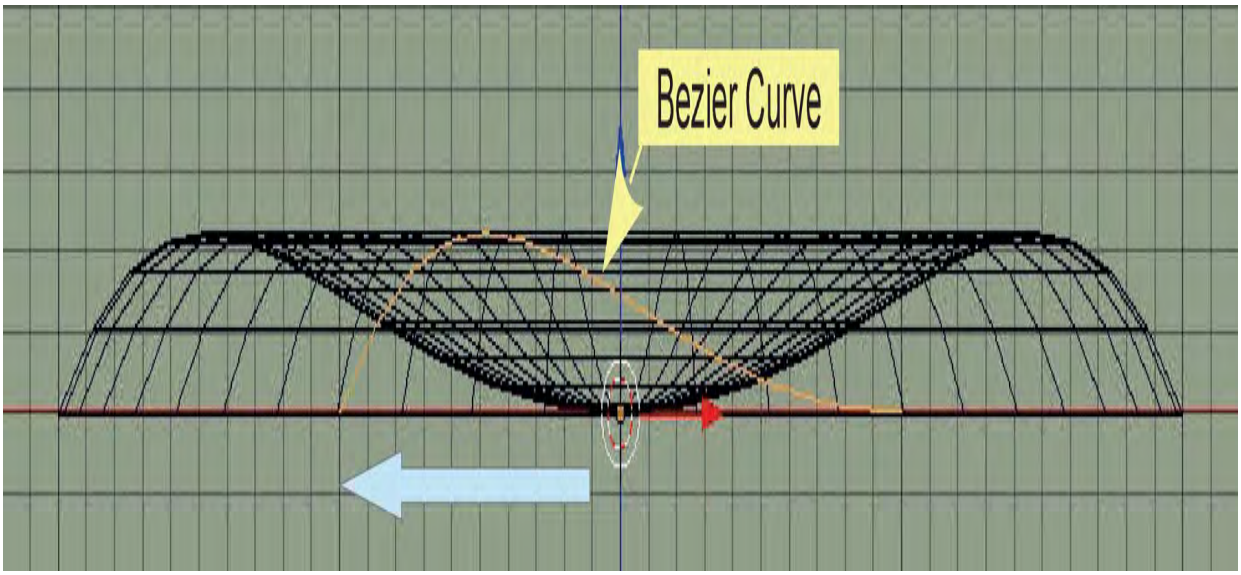


Figure 10.31

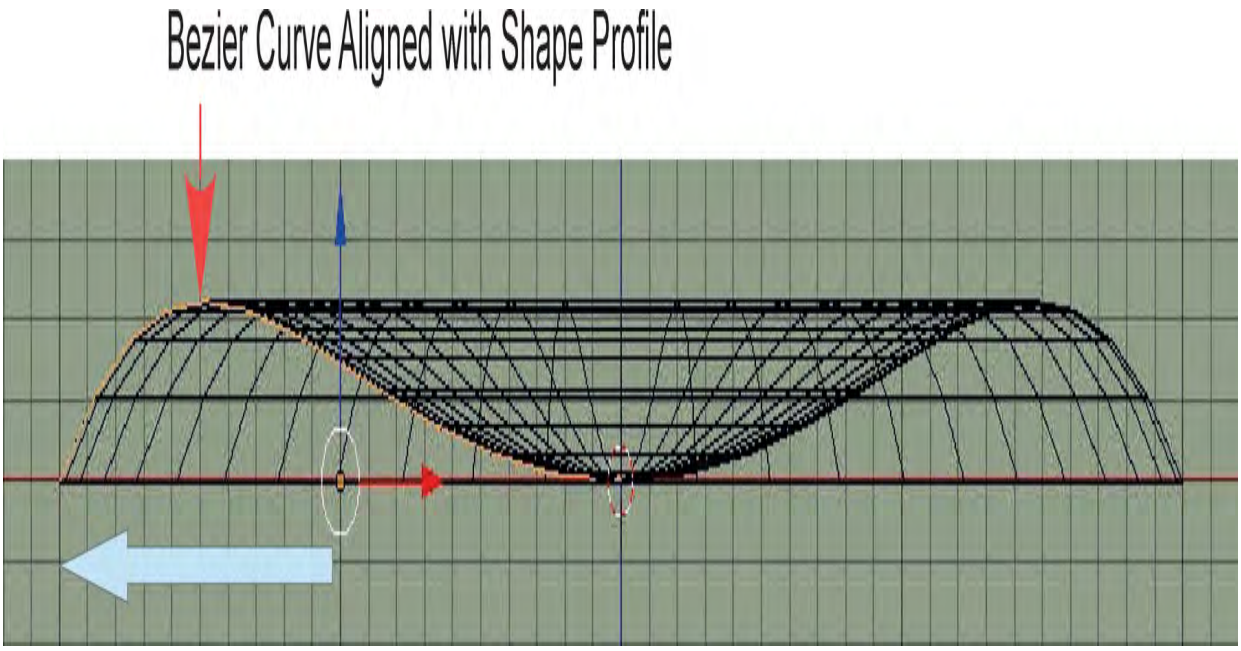


Figure 10.32

By skewing the 3D View Editor you will see that the shape has been generated by extruding the Curve profile through 360° (Figure 10.33).

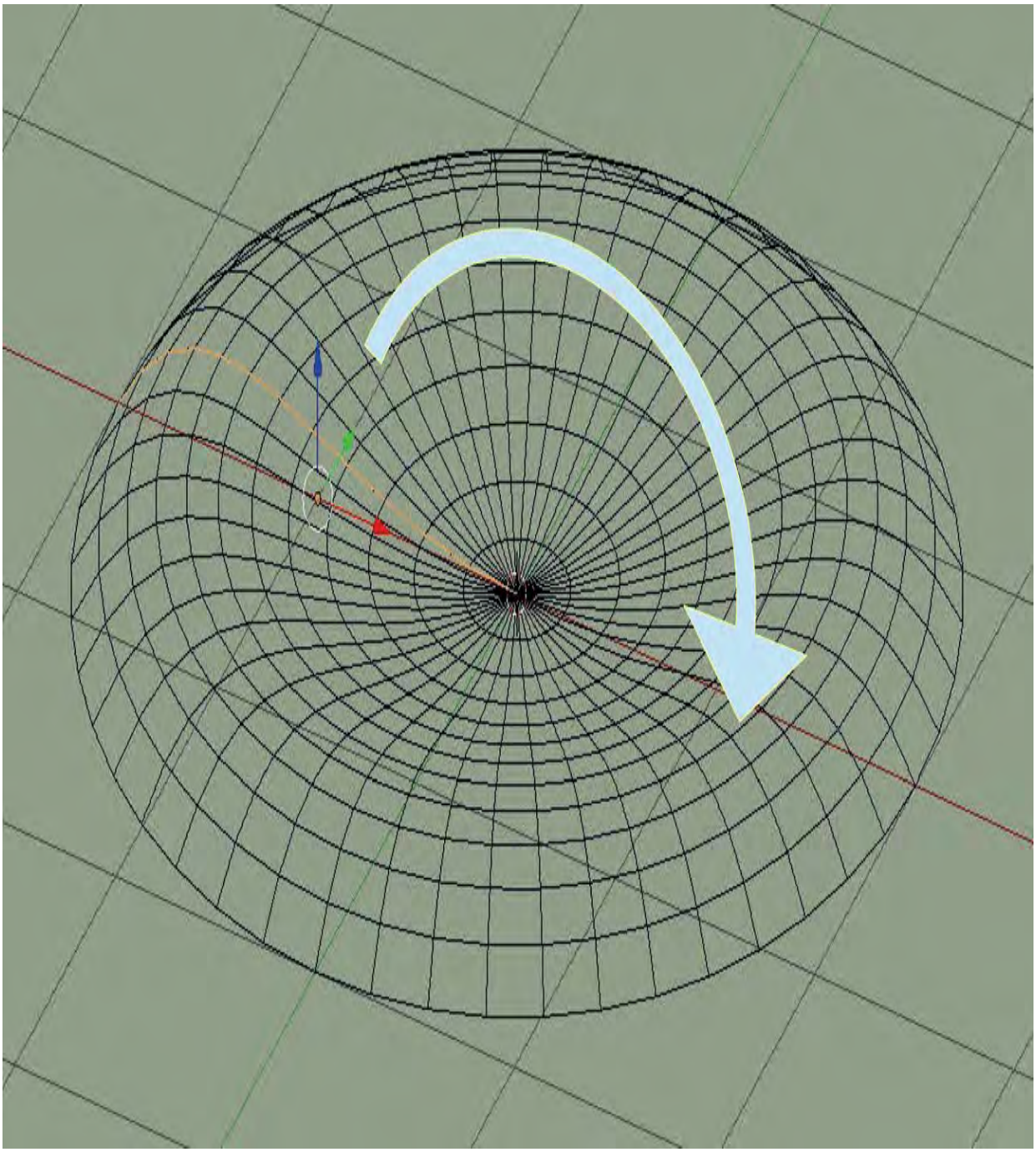


Figure 10.33

With the Bezier Curve still selected Tab to Edit Mode to see the **Control Handles** at each end of the Curve (Figure 10.33).

By selecting the Control Handle at the center of the shape and translating it along the **X Axis** towards the outside, you will see that it increases the inner

diameter of the shape. Similarly translating the Control Handle at the outer diameter alters the outer diameter (Figure 10.34).

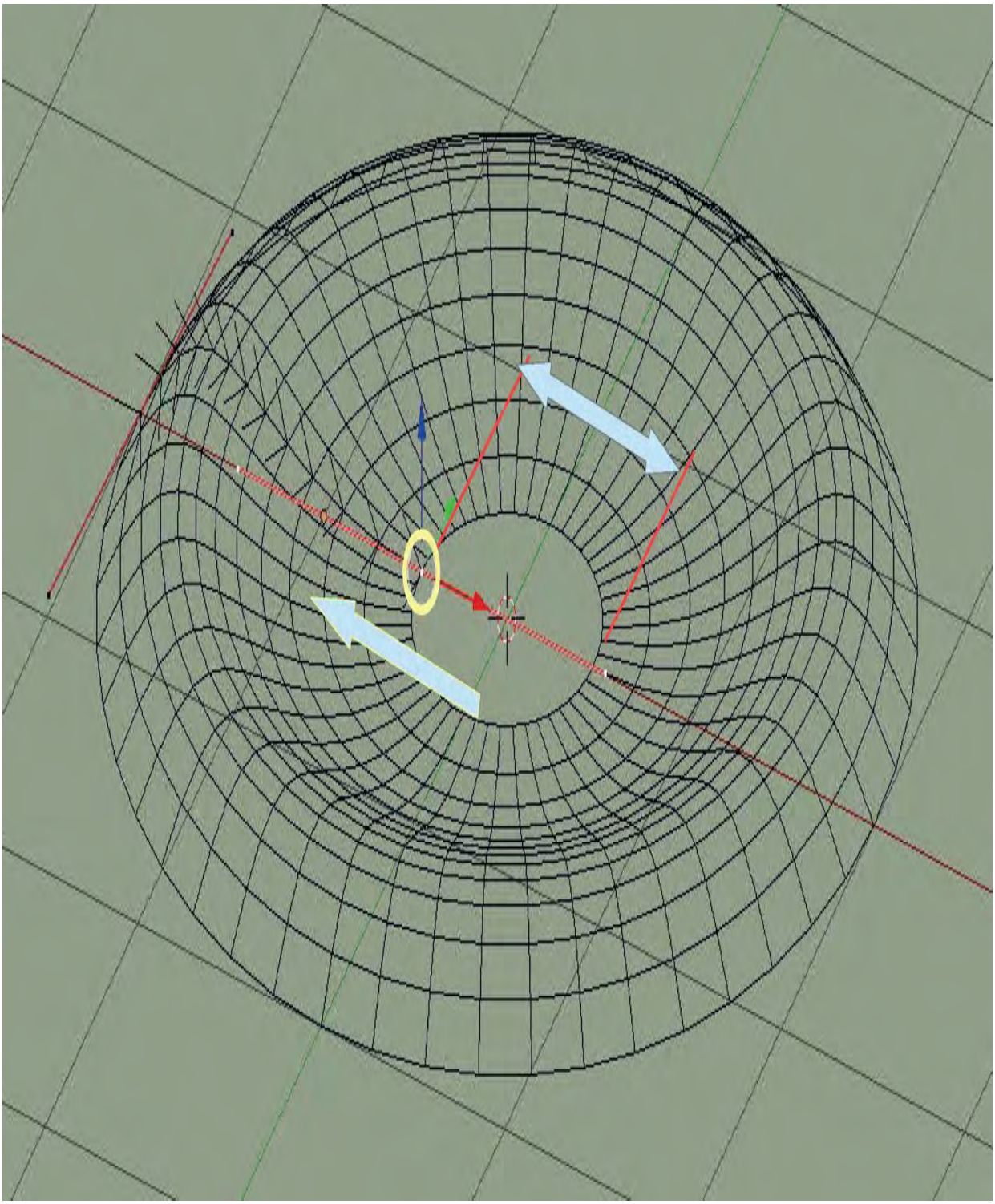


Figure 10.34

Shift select both Control Handles then press **R + Y + 90.** Doing this flips the Bezier Curve up on edge and changes the shape into something resembling a pot (Figure 10.35).

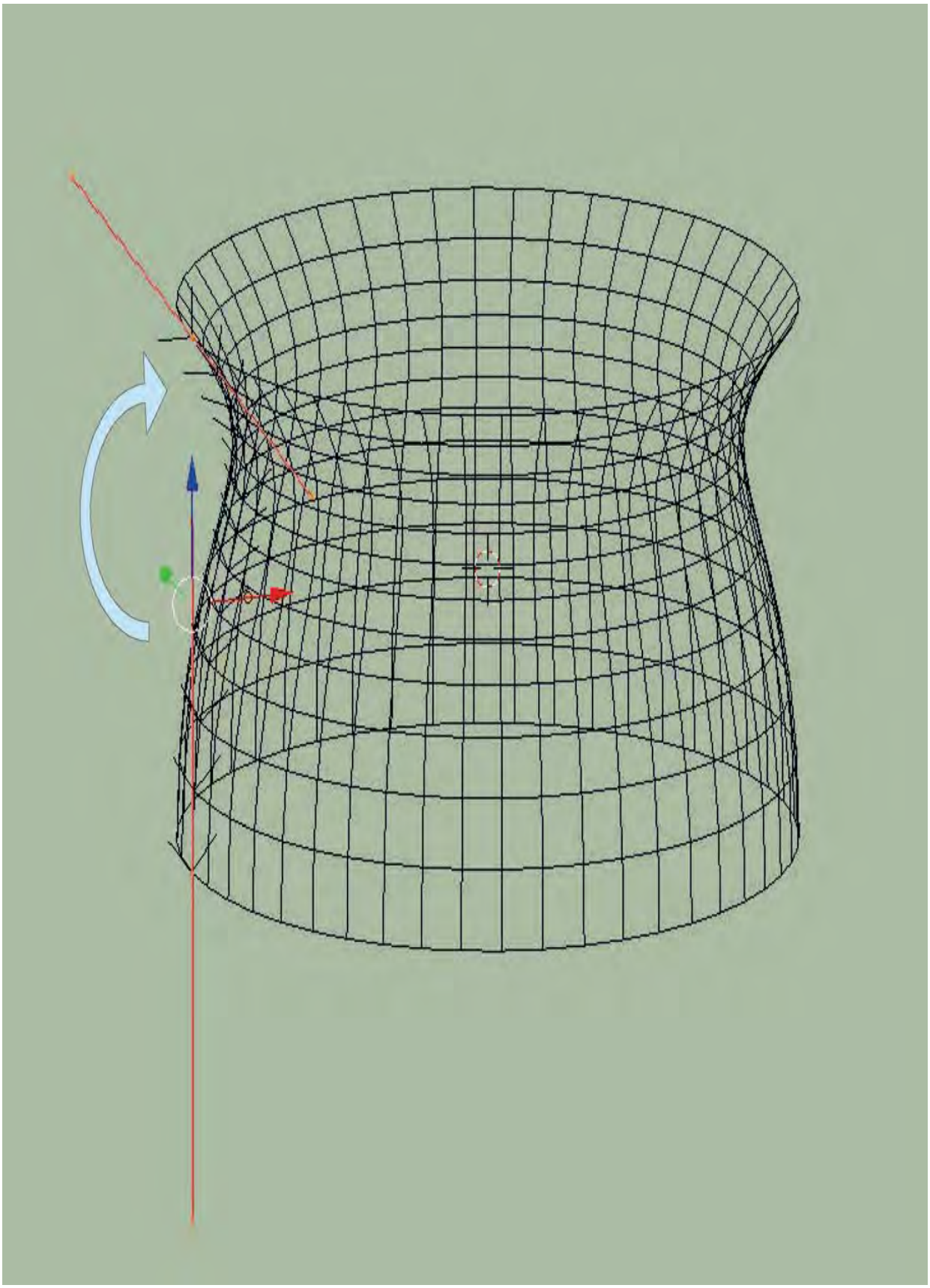


Figure 10.35

By manipulating the Control Handles you can modify the shape. Selecting both Control Handles and Subdividing (Click RMB in the Editor and select Subdivide in the Curve Context Menu) adds a third Control Handle (Figure 10.36). With the Control Handle selected, press the **V Key** to display the Handle type menu. Type **Vector** allows you to produce sharp corners when the handle is rotated.

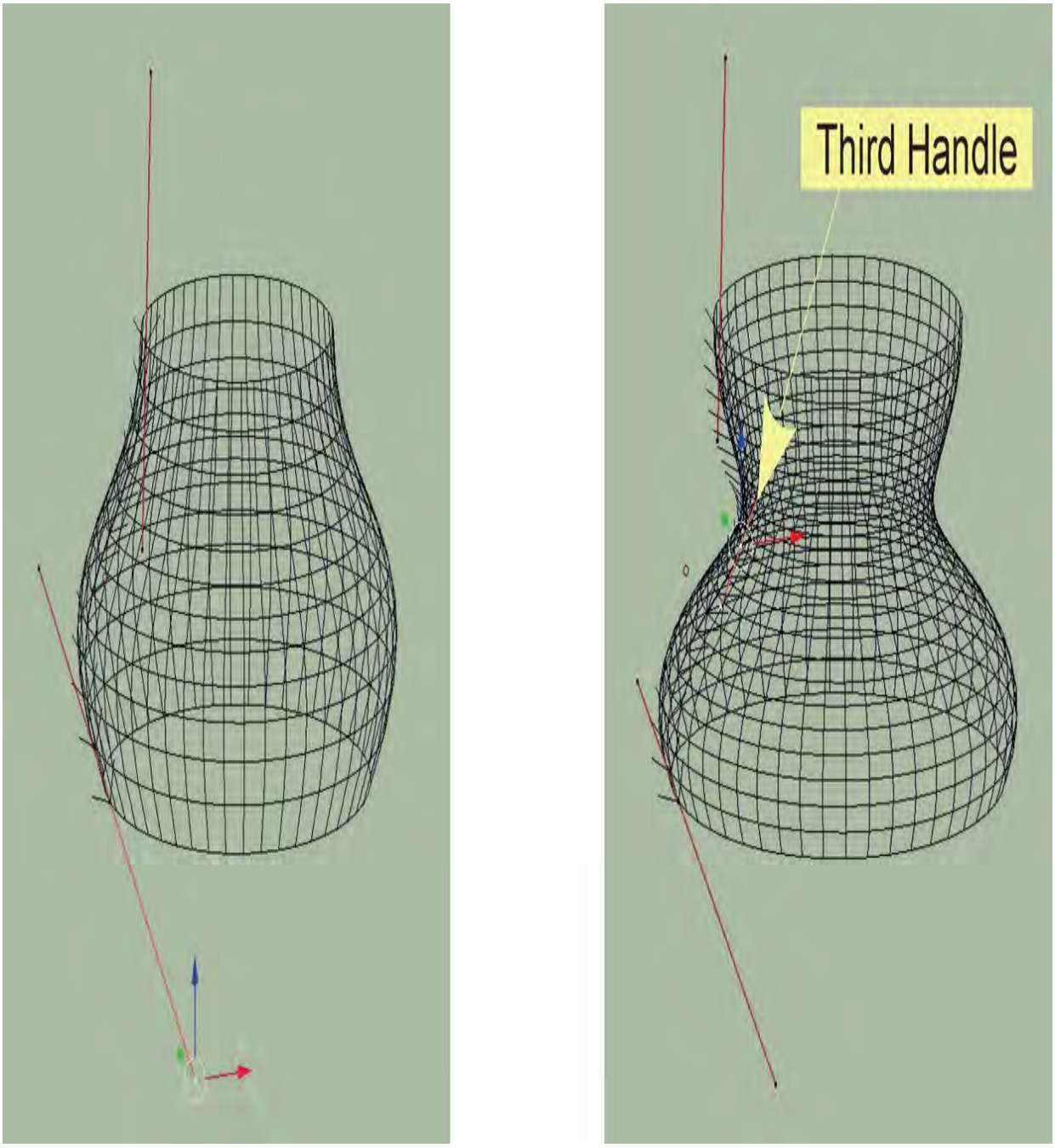


Figure 10.36

When you have completed shaping, Tab to Object Mode, deselect the **Bezier Curve** and select the shape itself. Change to **Solid Viewport Shading Mode** (Figure 10.37).

In the **Properties Editor, Object Properties (NOT Object Data), Viewport Display Tab** check (tick) **Wireframe** to view to see the subdivisions that will be created when the shape is converted to a Mesh Object (Figure 10.38). You may modify the subdivisions by altering values in the **Properties Editor, Object Data buttons, Shape Tab, Resolution.**

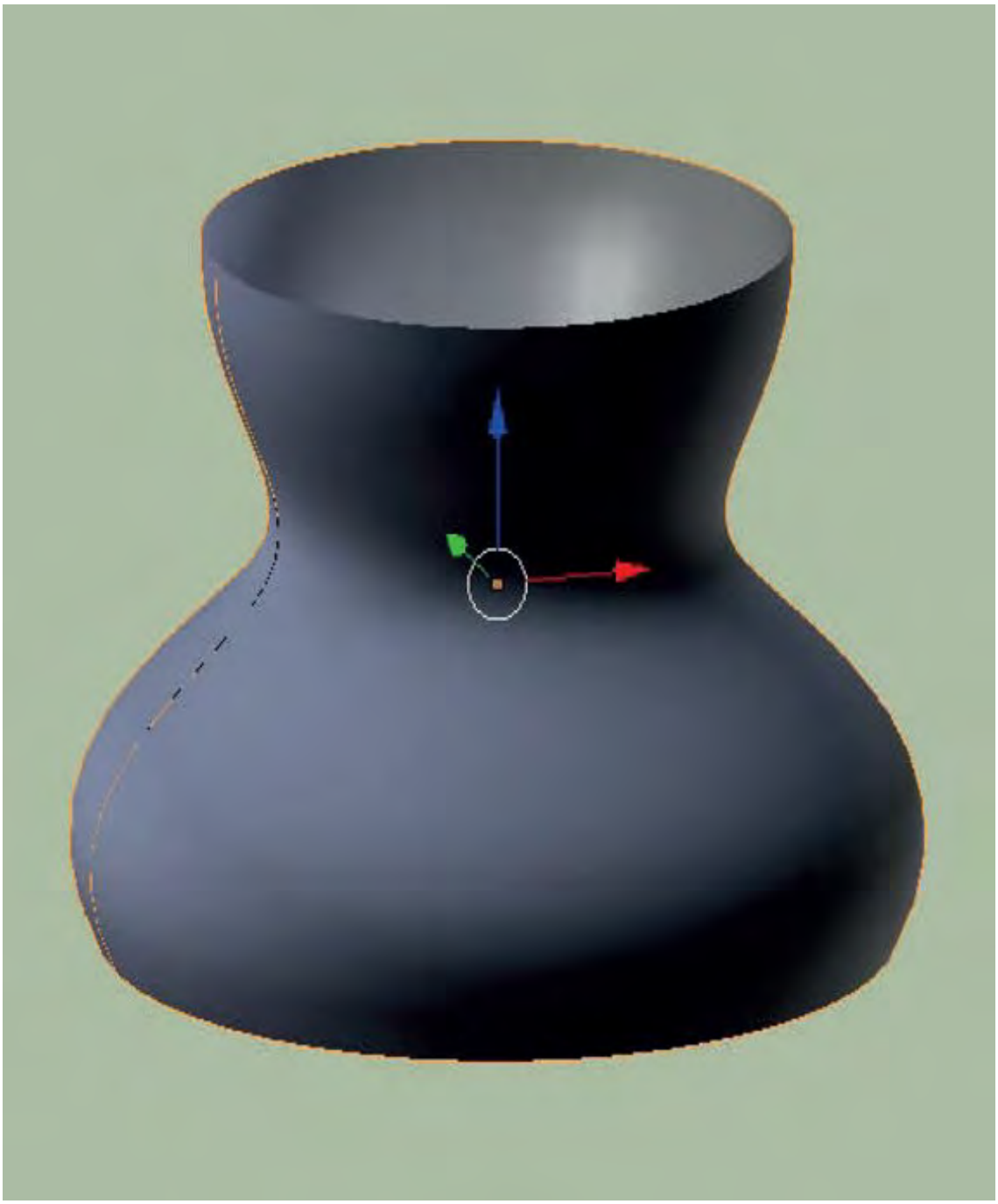


Figure 10.37

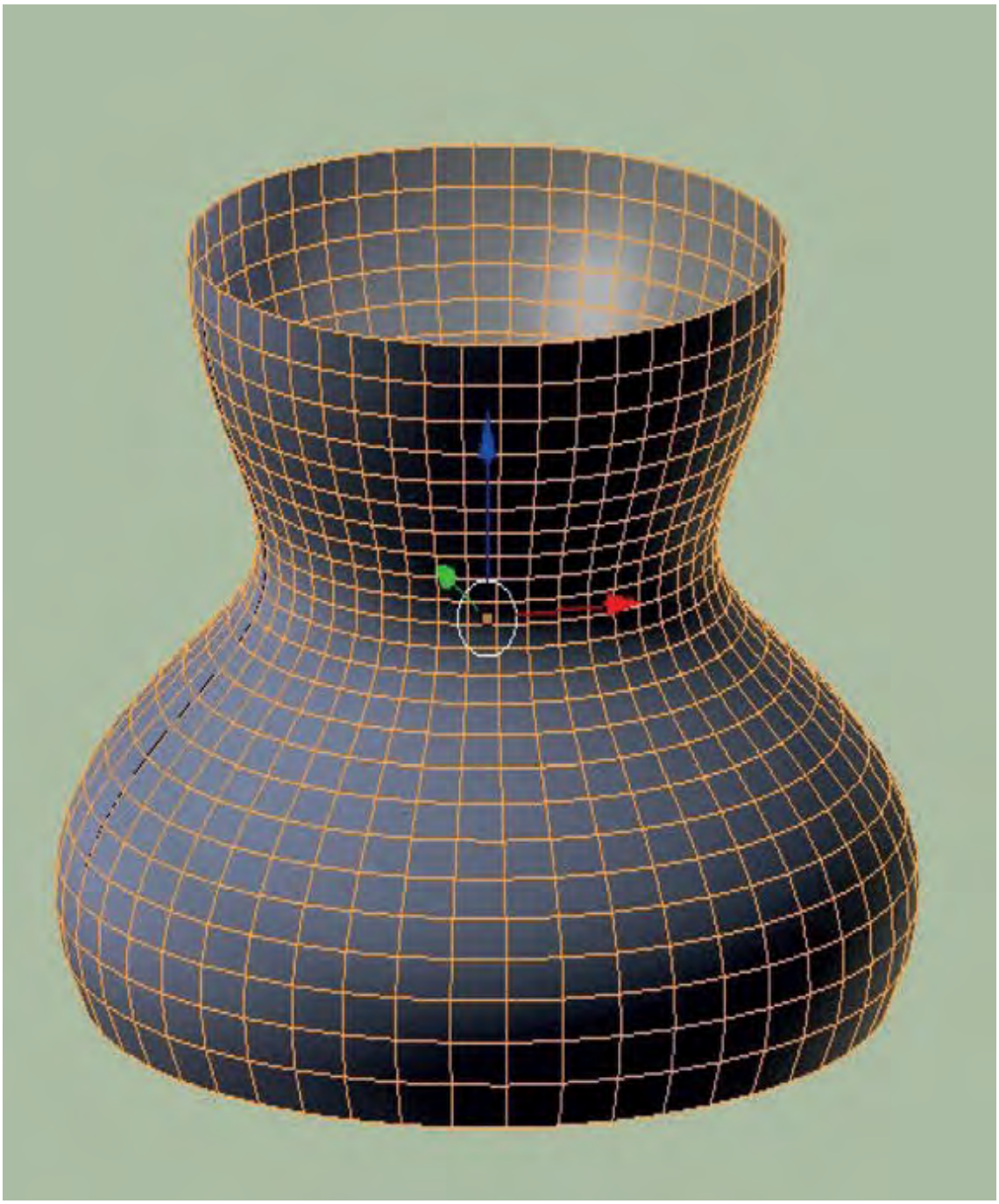


Figure 10.38

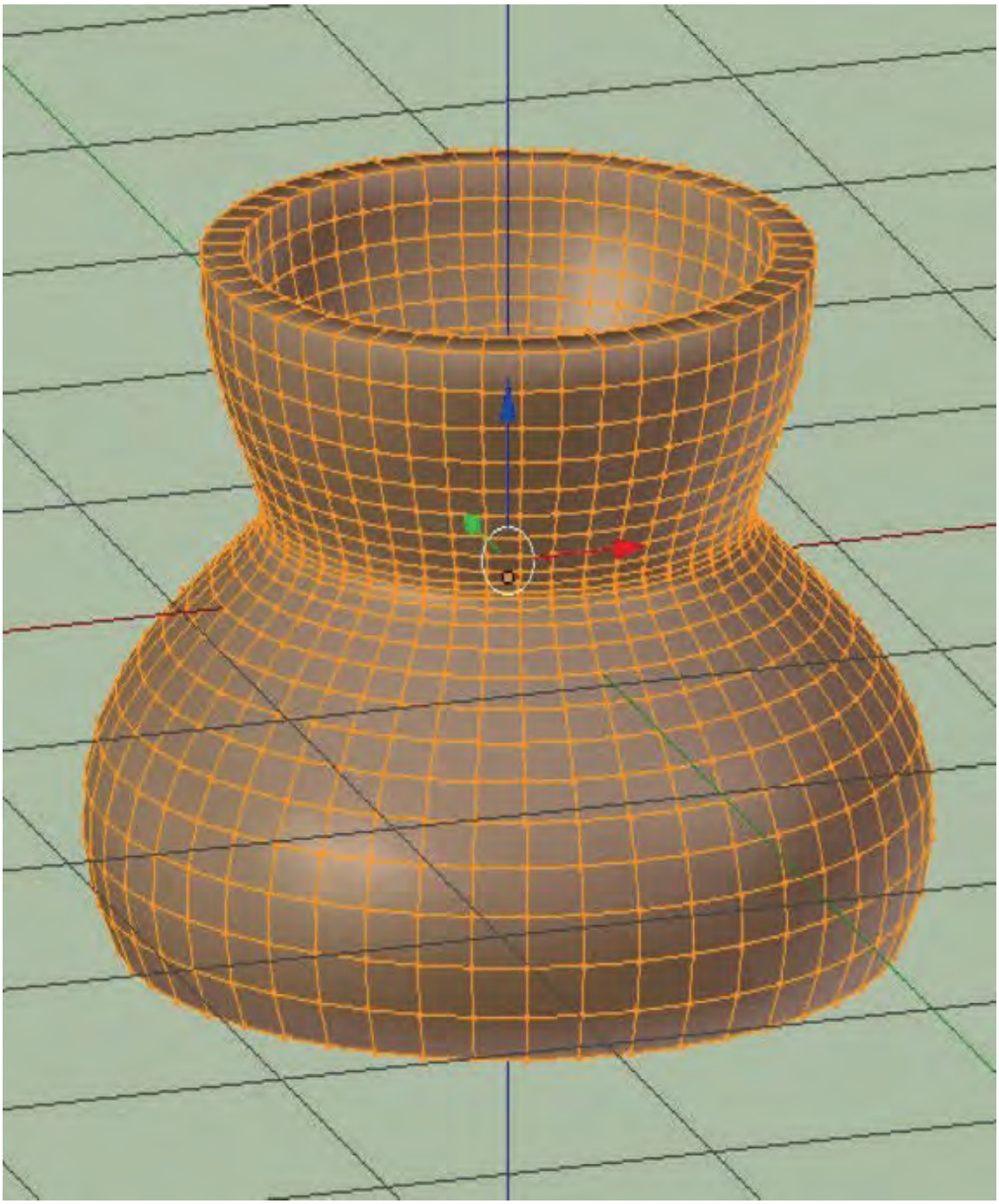


Figure 10.39

Finally, with the shape completed, click Object in the Header and select Convert to Mesh from Curve.

Add a **Solidify Modifier** (see Chapter 8 — 8.14) and increase the **Thickness value** to give the shape **wall thickness** (Figure 10.39).

11 Editing Techniques - Examples

11.1 Creating a Landscape
11.2 Dupliverts
11.3 Modeling an Aircraft
11.4 Sculpting - Sculpt Mode
11.5 Sculpting Demonstration
11.6 Creating a Humanoid Figure

Introduction

Becoming proficient at Modeling requires a knowledge of the tools that are available and how to combine the use of tools to create what you want. There are no hard and fast rules concerning which tool is used for any particular application. You use whatever suits what you are doing.

This chapter shows a very brief sample of how different tools are used in creating some basic models and effects.

What you can model using Blender is only limited by your imagination and your knowledge of what tools are available and where to find them.

11.1 Creating a Landscape - Proportional Vertex Editing

Proportional Vertex Editing can be employed to create a landscape or ground as a background for a Scene. Simply select Vertices on a subdivided Plane and move them up or down the Z Axis. Vertices moved down forming depressions can be turned into lakes or rivers by positioning a second Plane below the original and giving it a different color (Chapter 4 - 4.18). When the landscape is formed in Edit Mode, Tab to Object Mode, select the ground and

select **Shading Smooth** in the **Object Context Menu** (RMB click in the Editor).

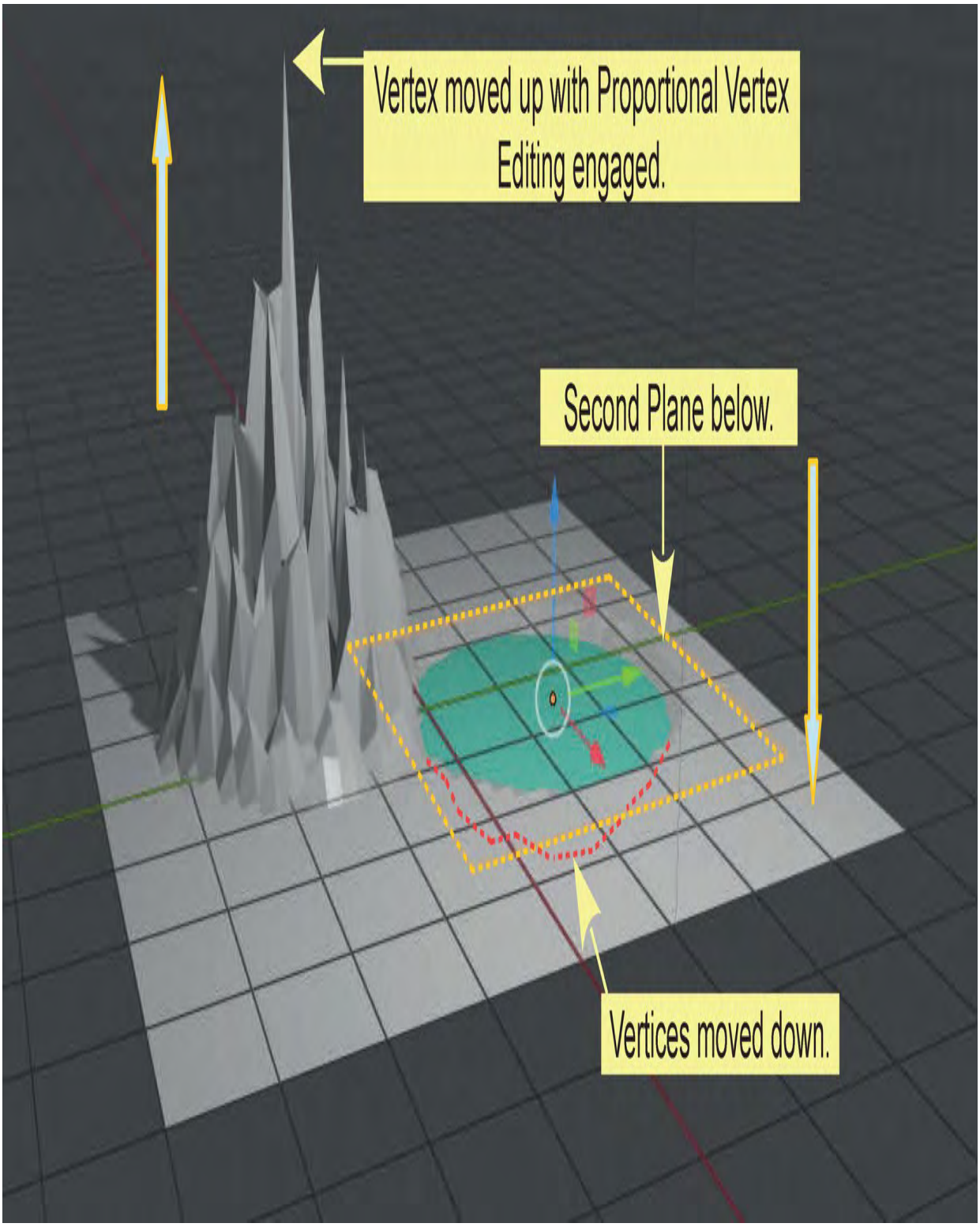

Vertex moved up with Proportional Vertex Editing engaged.
Second Plane below.
Vertices moved down.

Figure 11.1

Note: The Add Mesh: A.N.T. Landscape Add-on for Blender automates landscape generation. There are also external programs.

Cook Book Instructions

Delete the Cube from the default 3D Viewport Editor.

Add a Plane Object and scale up four times.

Tab to Edit Mode.

Subdivide the Plane.

In the Last Operator Panel increase the Number of Cuts to 10.

Subdivide again. Increase the Number of Cuts to 3(10 + 3 = 13).

Deselect the Vertices and select a single Vertex.

Enable Proportional Editing and select Random Falloff.

Press G Key plus Z Key and drag the Mouse Cursor up.

Deselect the Vertex and select a second Vertex. Repeat the process for another mountain or with

Spherical Fall off drag down creating a depression for a lake.

Add a second Plane to the Scene just below the first scaled to sit in the depression.

Give the new Plane a geen-blue color.

Note: When following these instructions, at this point, you will not see the view of the Scene in the 3D Viewport Editor exactly as shown in the diagram. What is seen in the Editor is dependent on how Material (Color) is applied (Chapter 16), what lighting (Lamps) have been introduced (Chapter 15) and the Viewport Shading that is implemented (Chapter 14).

11.2 Dupliverts

Dupliverts means **Duplicating at Vertices,** which means creating an Array of Objects by duplication. Each duplicated Object is positioned at the location

of a Vertices of a secondary Object. To demonstrate, a **UV Sphere** Object will be duplicated at the position of each Vertex of a **Plane** that has been Subdivided.

Delete the default **Cube** Object in the default Blender Scene. In the **3D Viewport Editor** add a **Plane** Object. Scale the Plane up four times (S Key + 4, LMB click) then **Tab** to **Edit Mode** and **Subdivide.**

To replicate the Subdivision in the diagrams, in 3D View Editor Header, RMB click in the Editor and select Subdivide in the Object Context menu. In the Subdivide Last Operator Panel increase the Number of Cuts to 10. Press the A Key to select all Vertices. RMB click again, Subdivide a second time. In the Last Operator Panel the Number of Cuts resets to 1. Increase to 2. If you like to examine the subdivision in detail you will find you have 1156 Vertices.

How many times you Subdivide is arbitrary but having a decent number of Vertices produces a good effect.

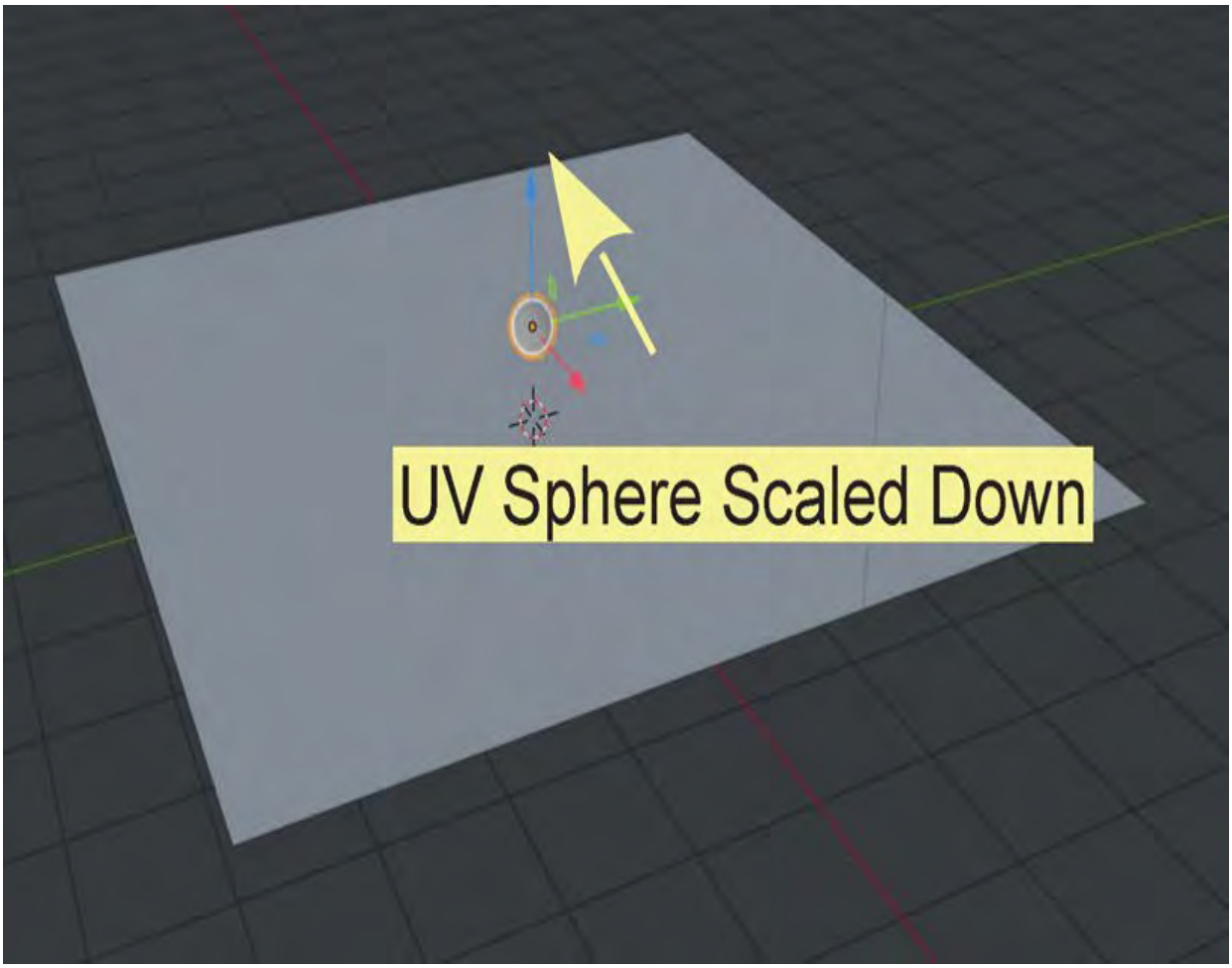


Figure 11.2

Tab to **Object Mode** deselect the Plane. Add a **UV Sphere** Object to the Scene, Scale it down (Figure 11.2). In the Sidebar Object Properties Panel (N Key) make the Scale values = 0.158.

With the **Sphere selected** (in Object Mode), **Shift** select the **Plane.** Press **CTRL + P key** and select **Object** to parent the Sphere to the Plane (Figure 11.3).

Deselect the Sphere and the Plane. Select the Plane only.

Figure 11.3

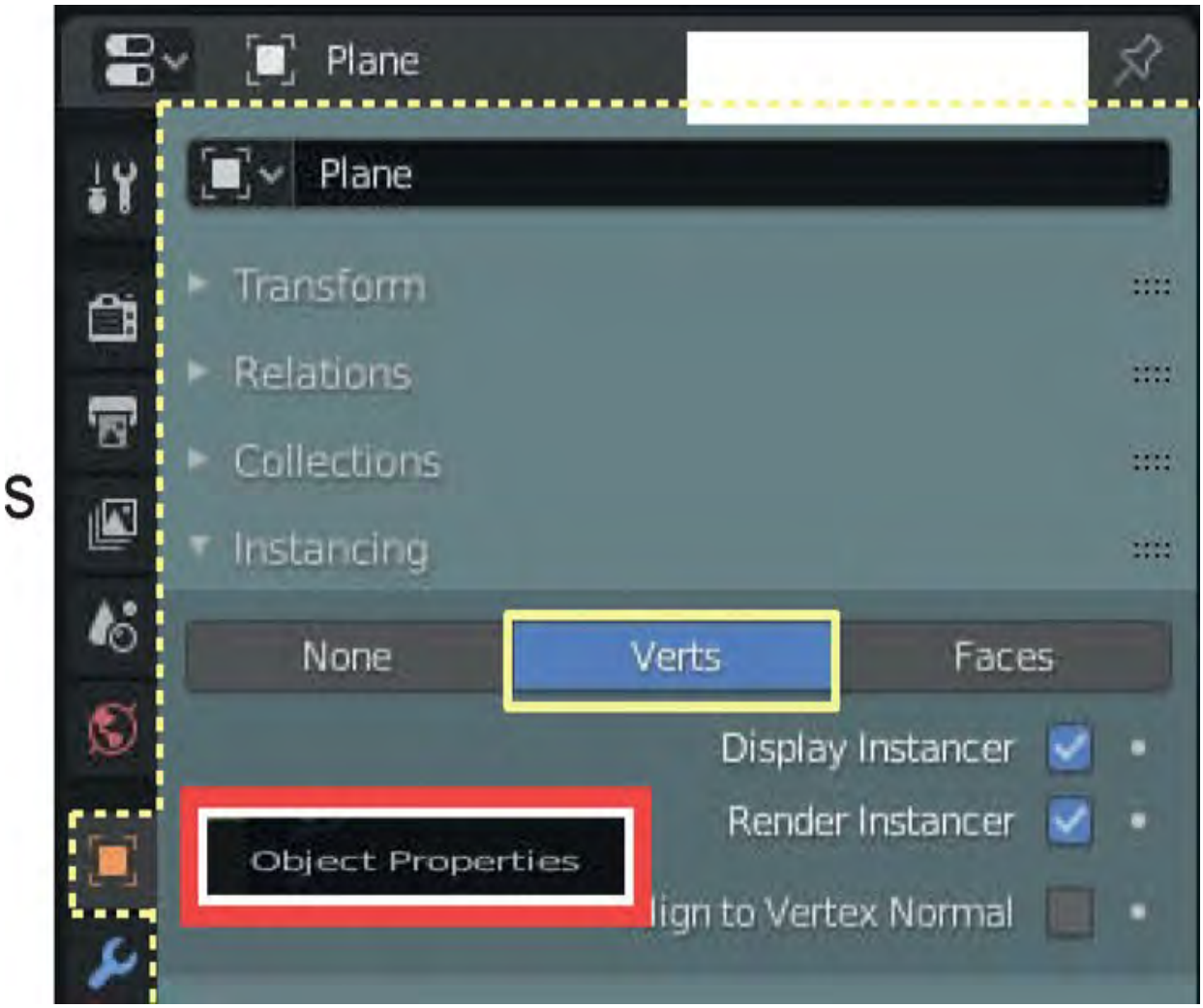


Figure 11.4

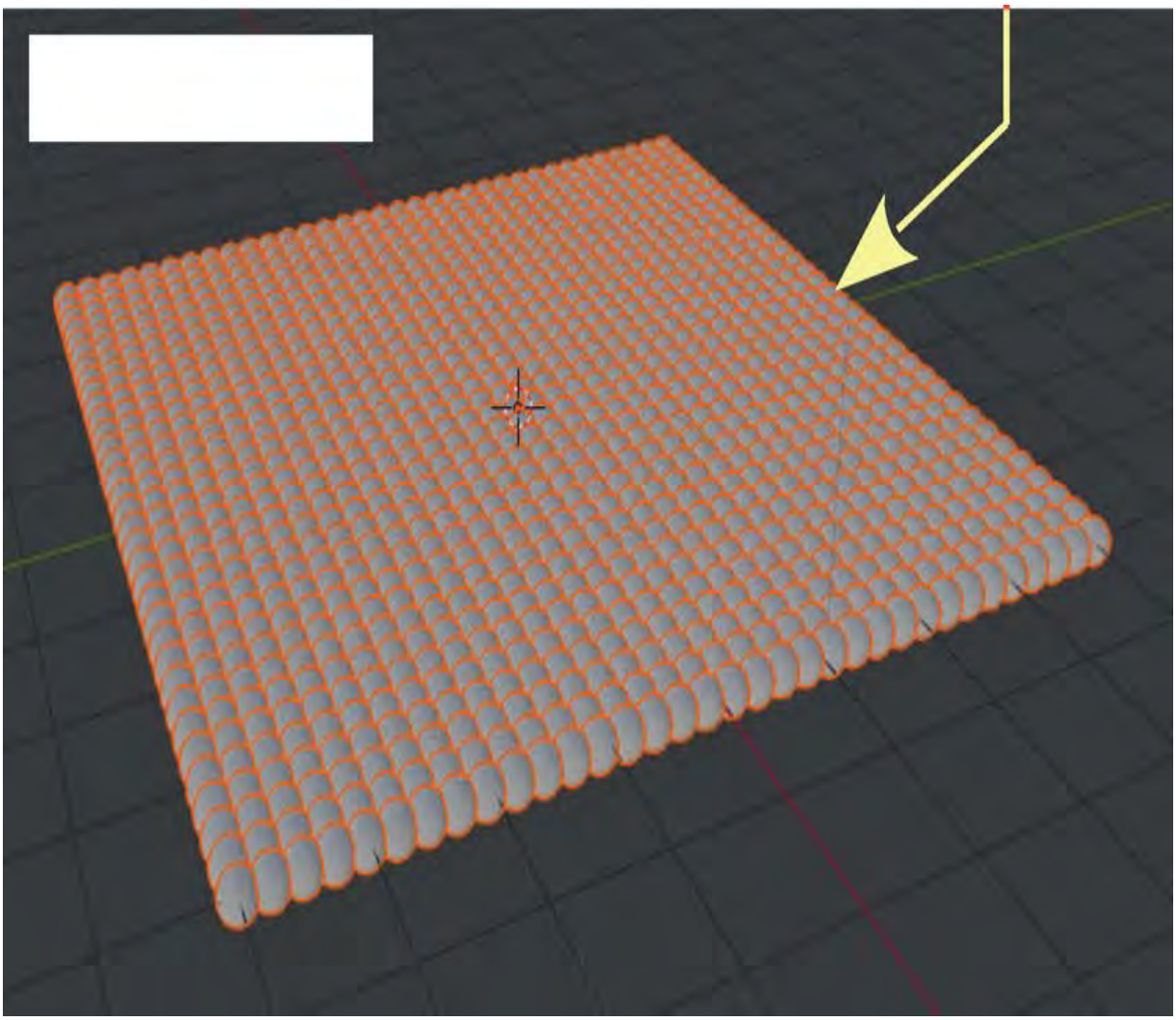


Figure 11.5

Deselect the Plane.

Figure 11.6

Scale the **Sphere** to adjust its size and create separation between the duplicates. Since the original Sphere is at the center of the Plane it is more than likely hidden among the duplicates and difficult (impossible) to select.

To select the Sphere, go to the **Outliner Editor** and locate the Sphere in the File tree (Figure 11.6). Select the Sphere in the **Outliner Editor** (Click LMB) then in the **3D Viewport Editor** Scale the Sphere down to make gaps between the duplicates (Figure 11.7).

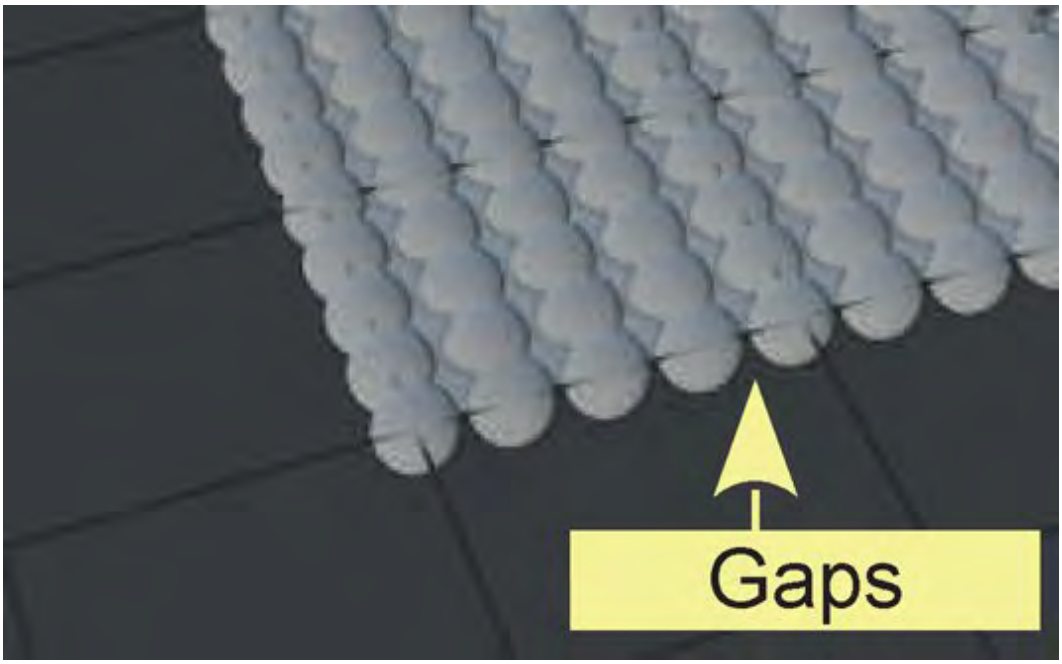


Figure 11.7

With Sphere selected, in **Object Mode,** Move up on the Z Axis (Figure 11.8). Deselect the Sphere.

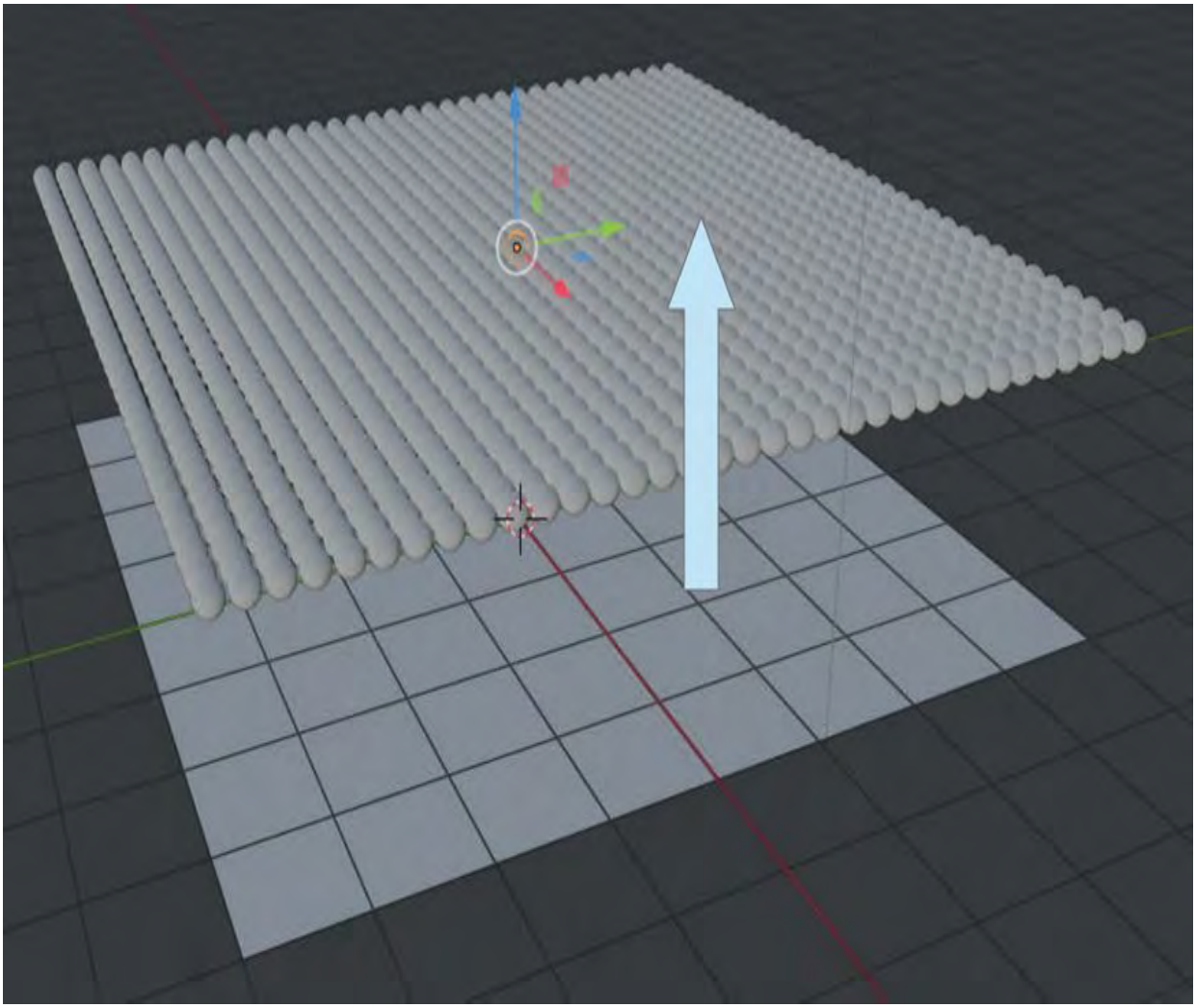


Figure 11.8

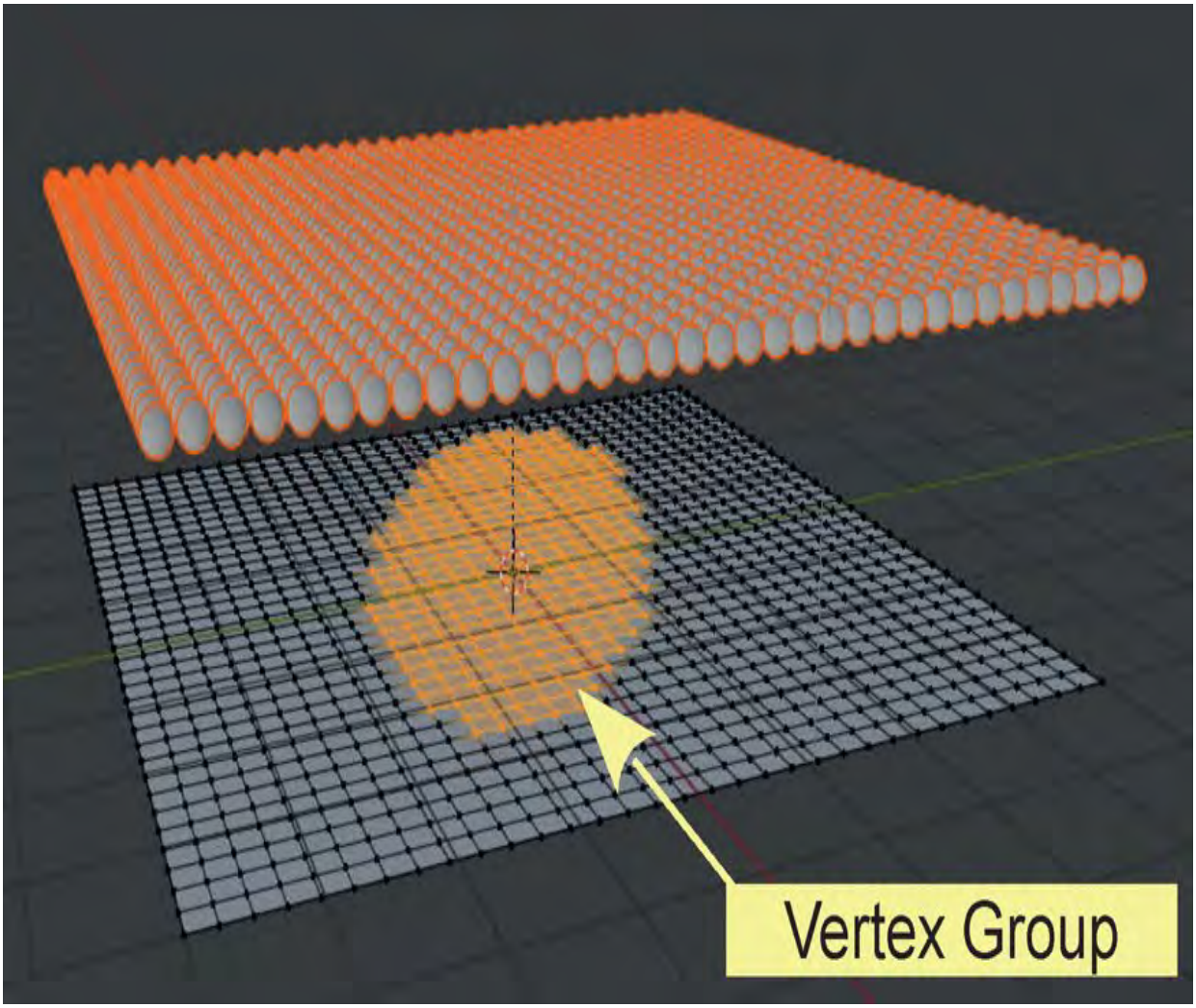


Figure 11.9

Select the **Plane,** then in **Edit Mode** a group of Vertices may be selected (Figure 11.9) and with **Proportional Editing** turned on, translated deforming the Array of Spheres (Figure 11.10).

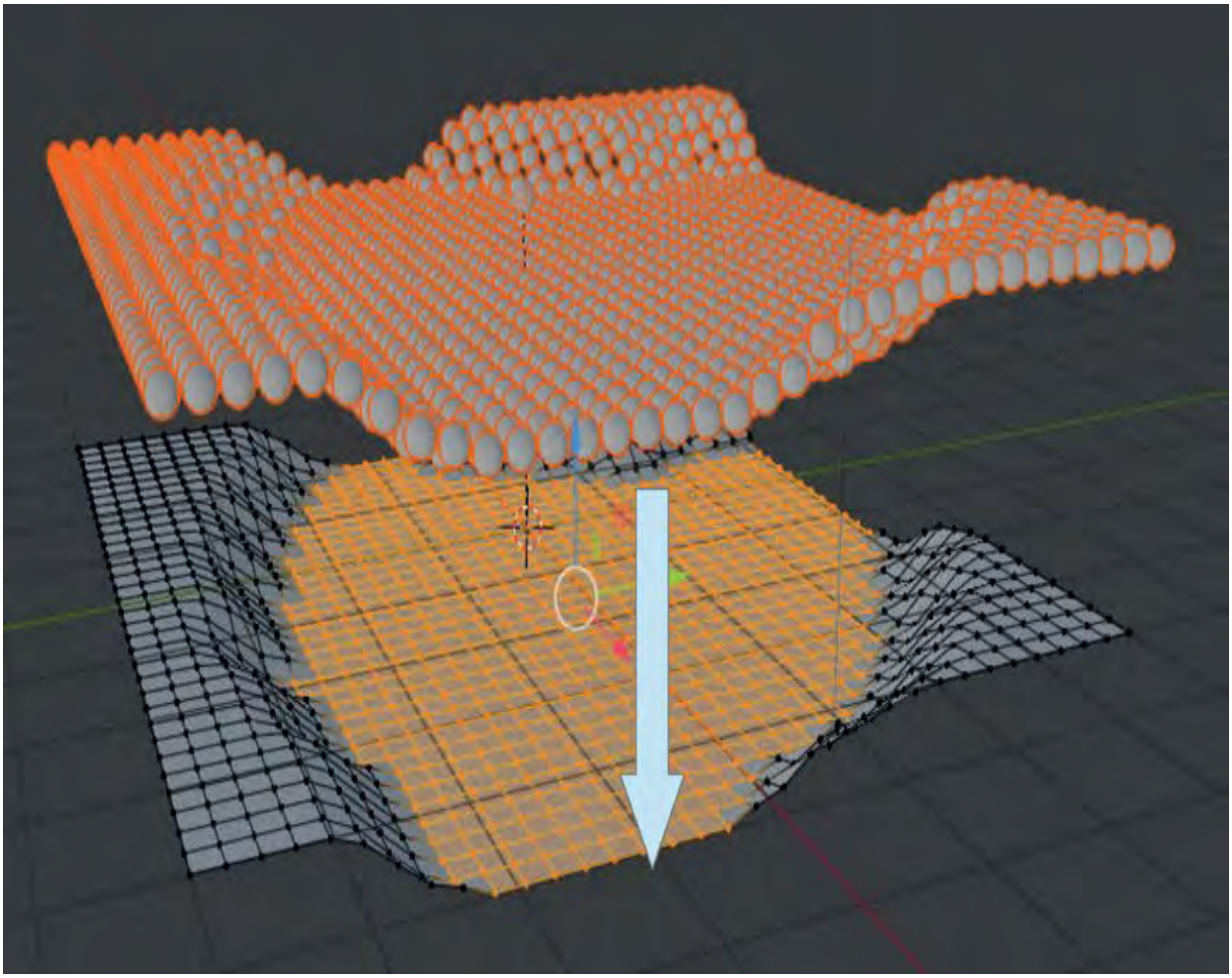


Figure 11.10

By animating the Vertices on the Plane to move and combining sound effects the Spheres can be made to dance in time to music.

11.3 Modeling Exercise—Aircraft

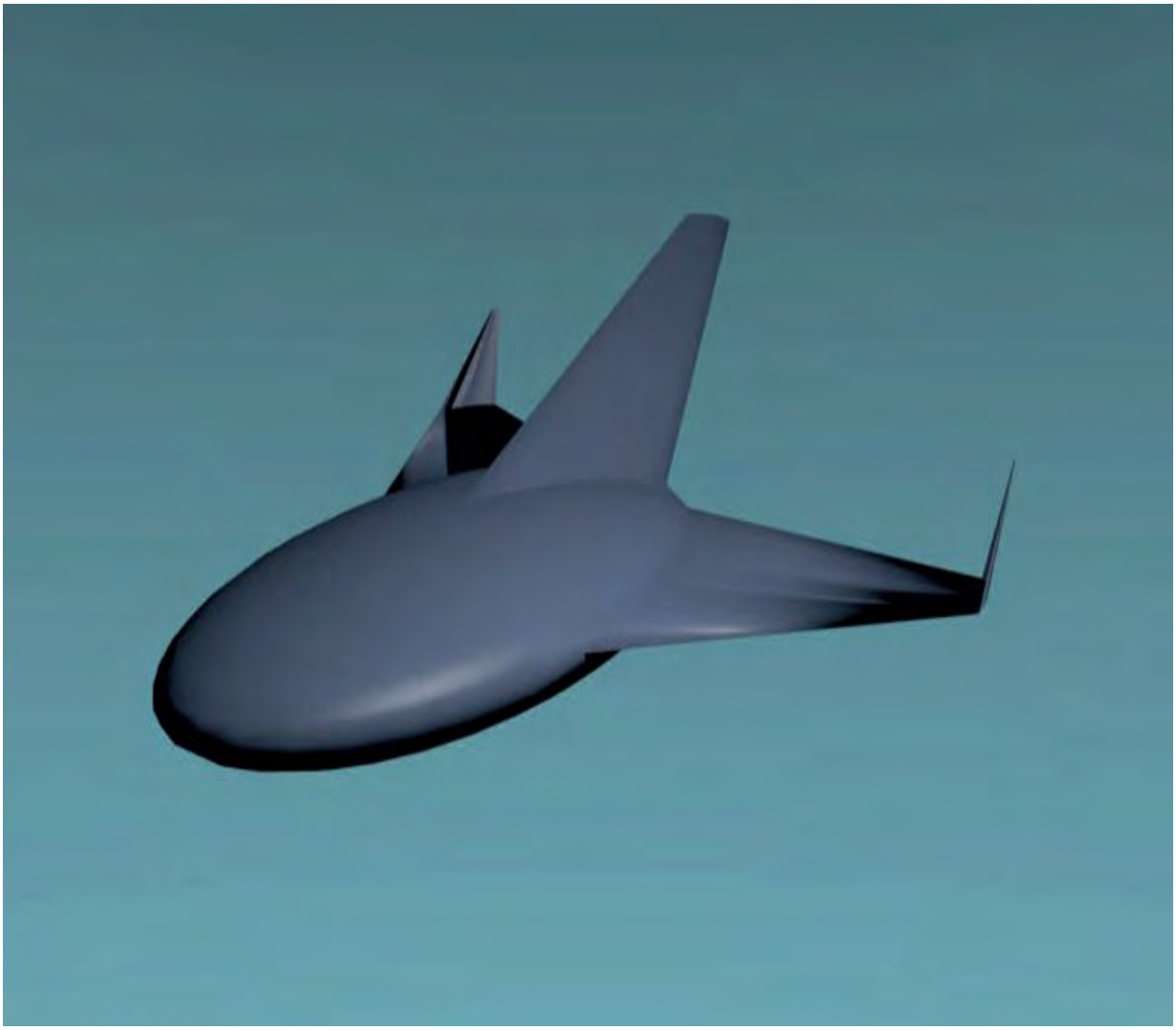


Figure 11.11

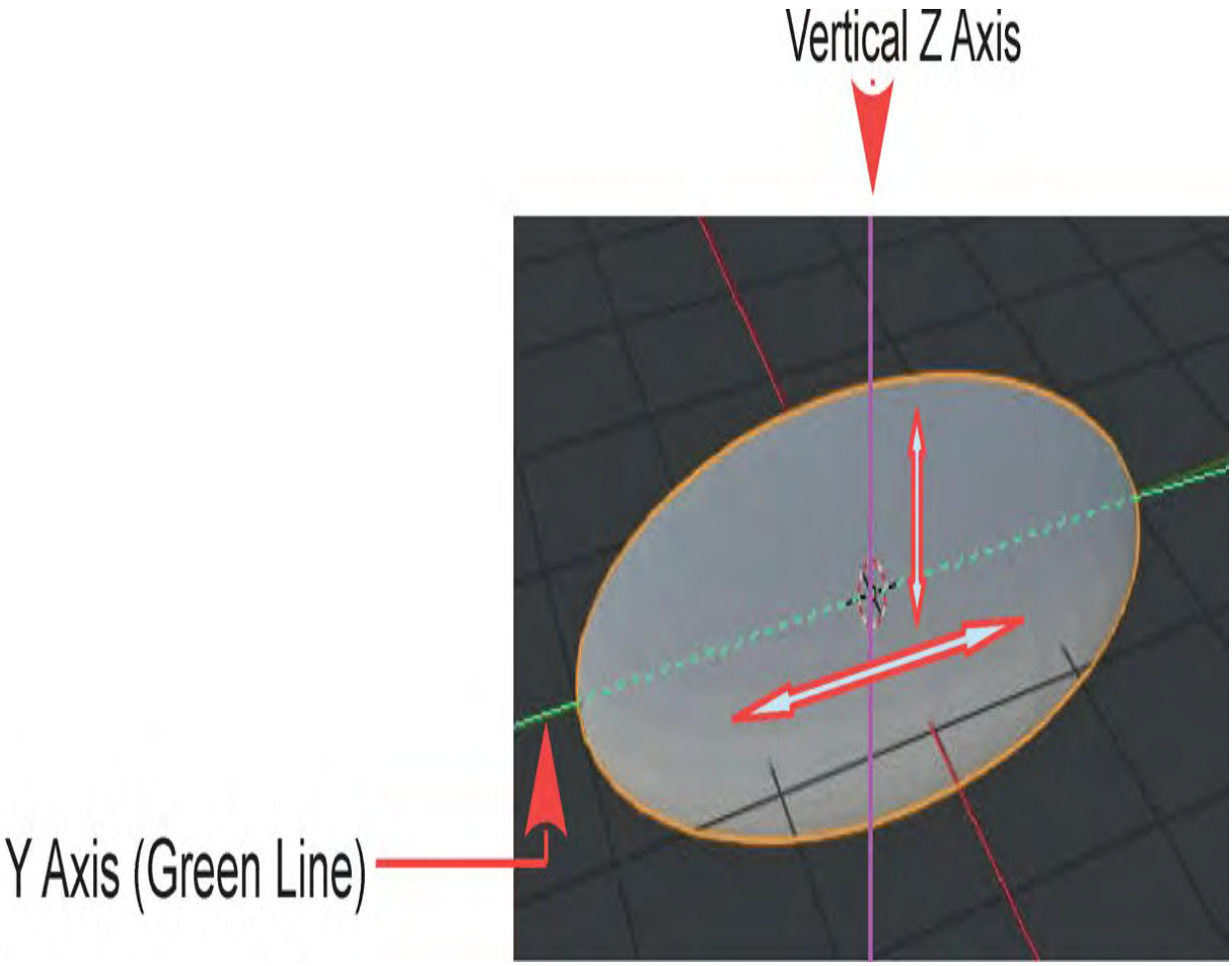


Figure 11.12

You will be modeling the reshaped Sphere and want it to be identical either side of the Y Axis (the green line). That is to say you want it to be mirrored **on the X Axis** (along the red line).

In Top Orthographic view, place the UV Sphere in **Edit Mode** and delete all the vertices on the LHS of the Y Axis (press the B key for Box select, place the Cursor as shown by the white cross, click and hold drag a rectangle around the LHS Vertices). **Don't forget; Activate Show X-Ray.** With the Vertices selected press the **X Key** and select **Delete Vertices .**

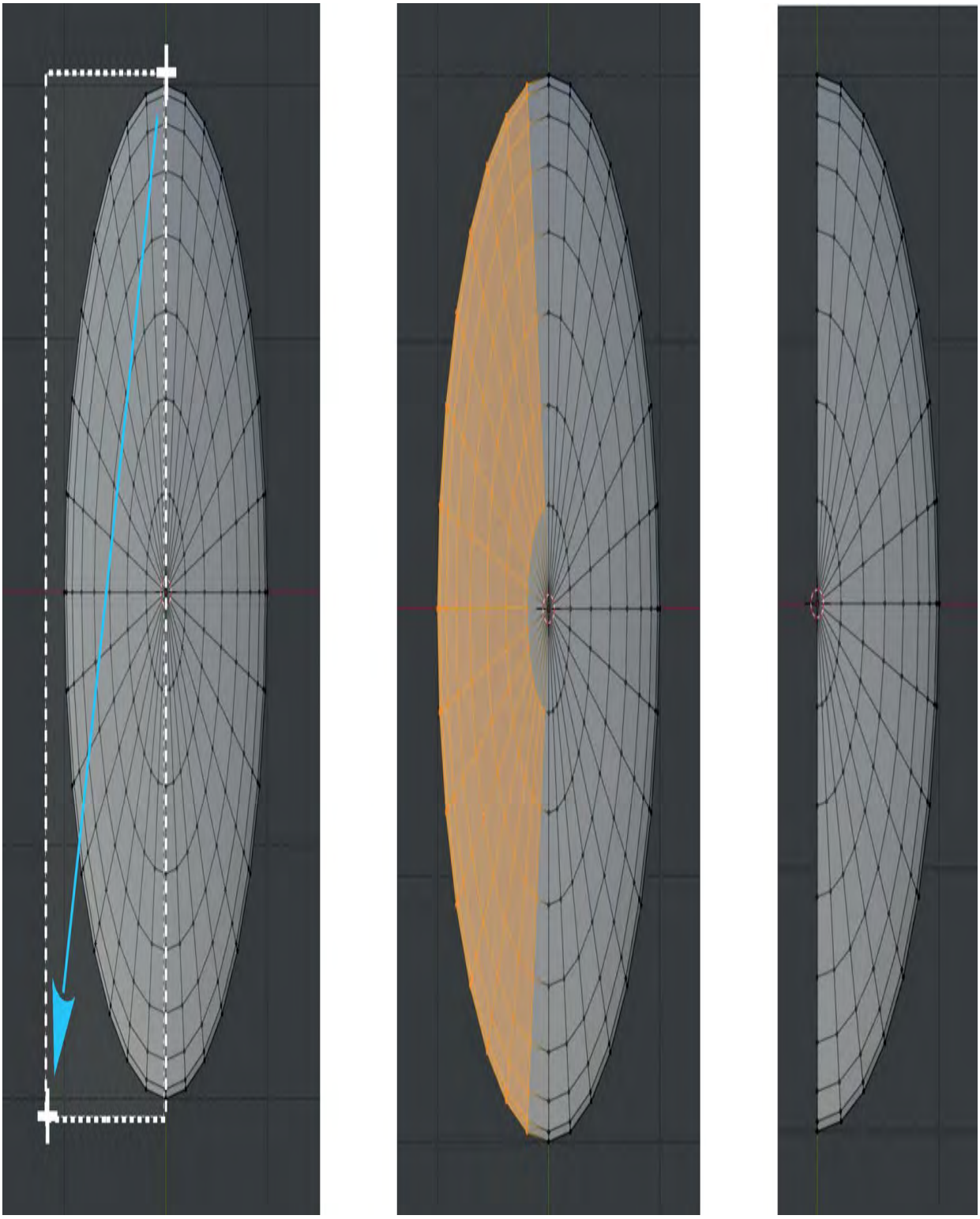


Figure 11.13

Place the UV Sphere in **Object Mode** (press Tab). Add a **Mirror Modifier** (Ref: Chapter 8 - 8.8) which by default is set to mirror on the X Axis. Change to **Right Orthographic** view (press the Num Pad 3 Key)

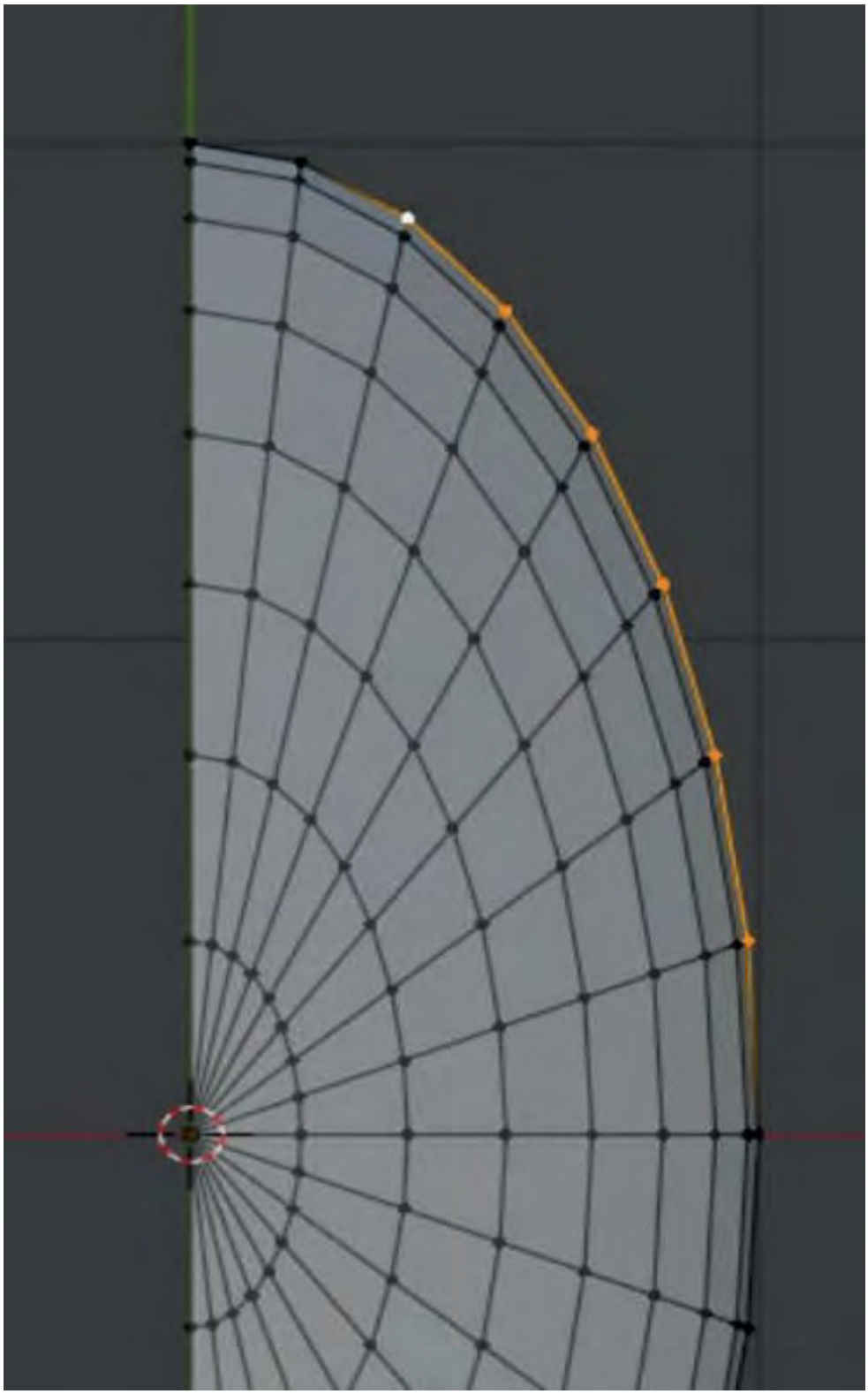


Figure 11.14

Tab to Edit Mode and select the Vertices as shown in Figure 11.14 (hold Shift and RMB click on each Vertex). Change back to **Top Orthographic** view. Press the **G Key** (Move) and use the **Widget** to move the Vertices to the right (Figure 11.15). With the **Vertex Group** still selected press the **S Key** (Scale) and Scale the group in.

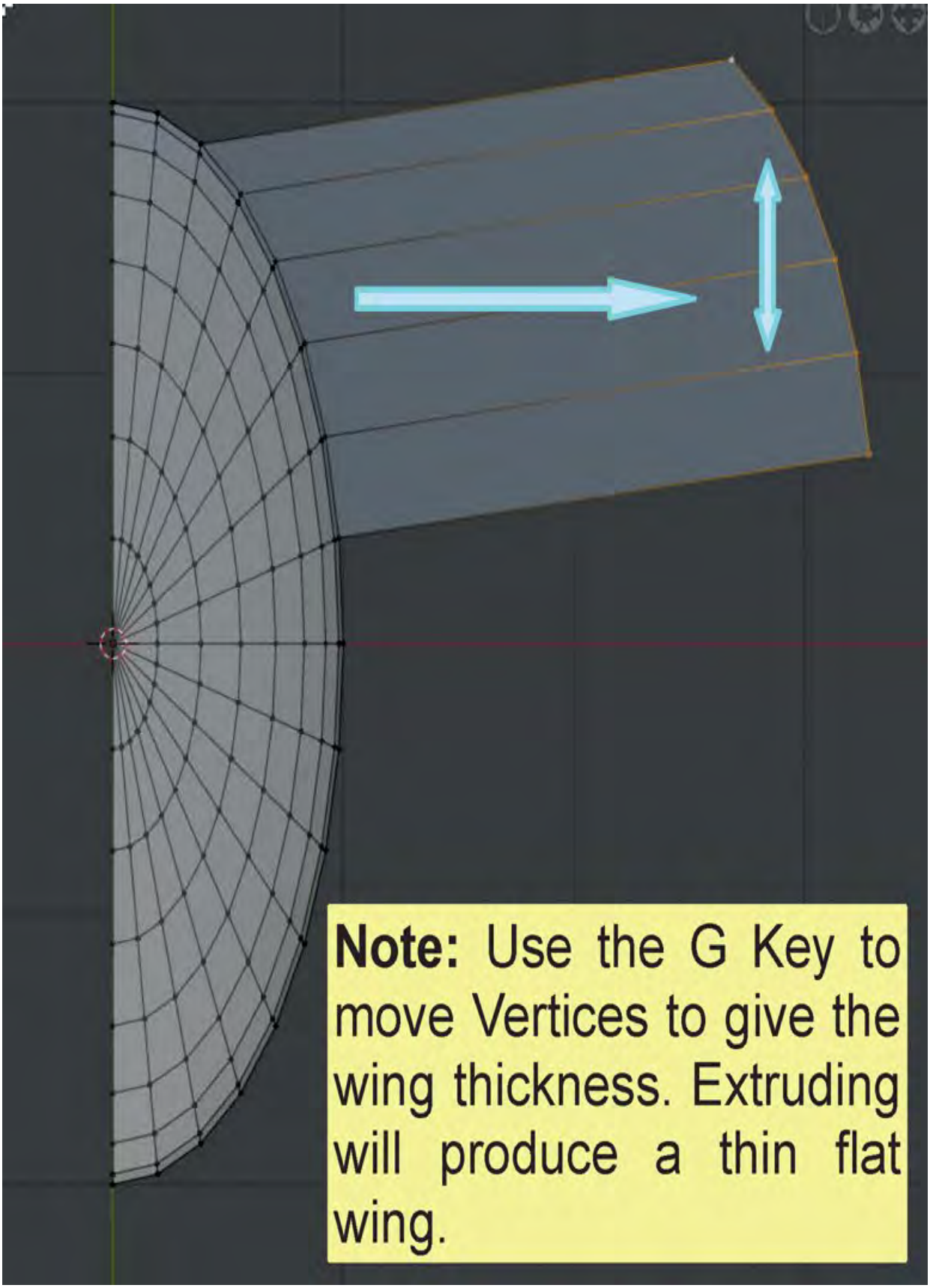

Note: Use the G Key to move Vertices to give the wing thickness. Extruding will produce a thin flat wing.

Figure 11.15

Use the **Widget** to move the group towards the back of the aircraft forming a wing (Figure 11.16).

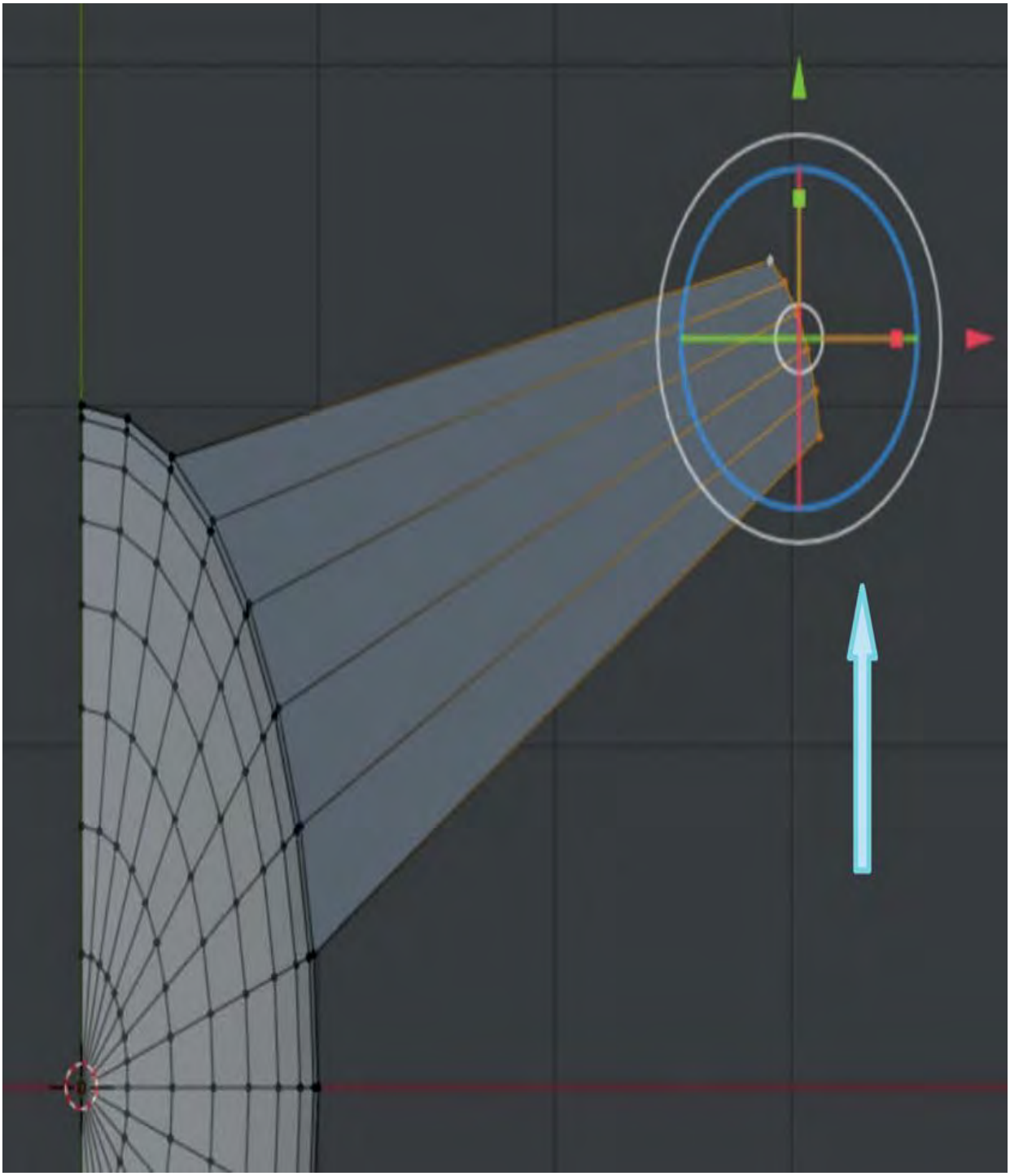


Figure 11.16

Note: Use the G Key to move Vertices to give the wing thickness. Extruding will produce a thin flat wing.

Rotate to align with the fuselage (Figure 11.16).

Extrude up, Scale the group in and move back forming the wing stabiliser (Figure 11.17)

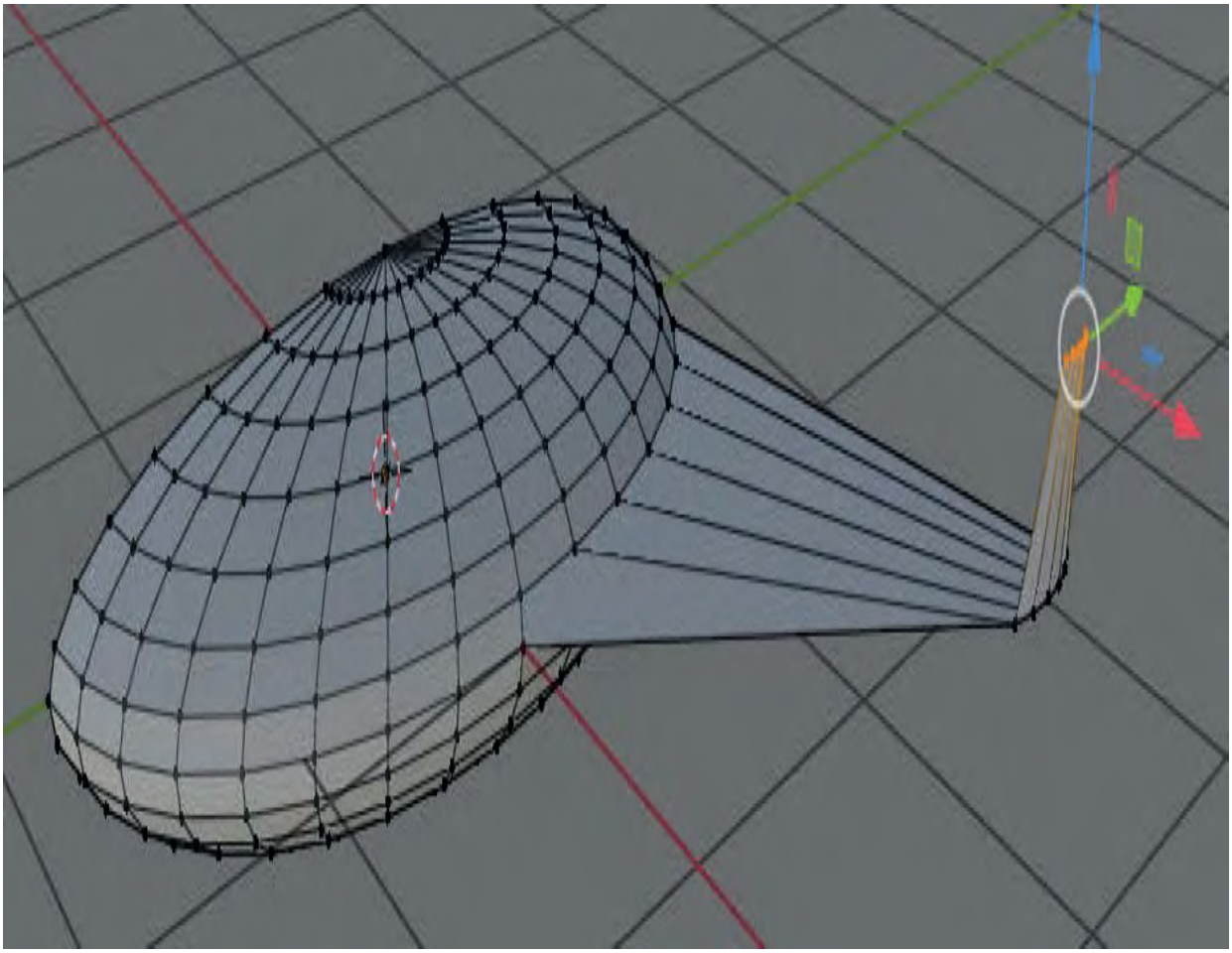


Figure 11.17

Select vertices at the rear of the fuselage on the centerline, Move up and repeat the procedure for the wing stabiliser forming a tail (Figure 11.18)

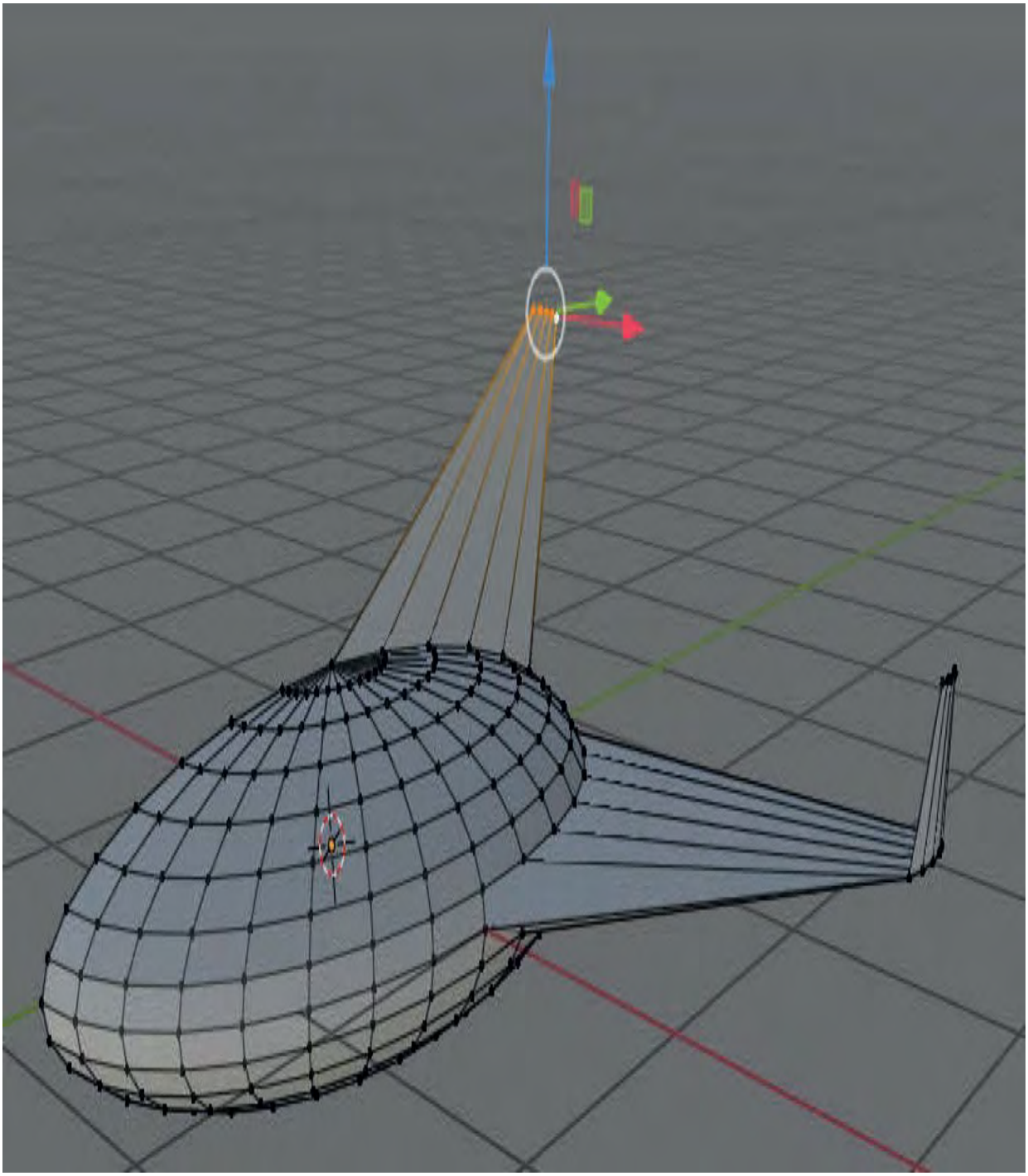


Figure 11.18

Go into **Object Mode, Apply the Mirror Modifier,** add a **Material.** In the **3D View Editor Header** click on **Object** and select **Shade Smooth.** Rotate the 3D View Editor to see your super duper aircraft.

Figure 11.19

Note: Change the 3D Viewport Editor background color to get the blue sky effect.

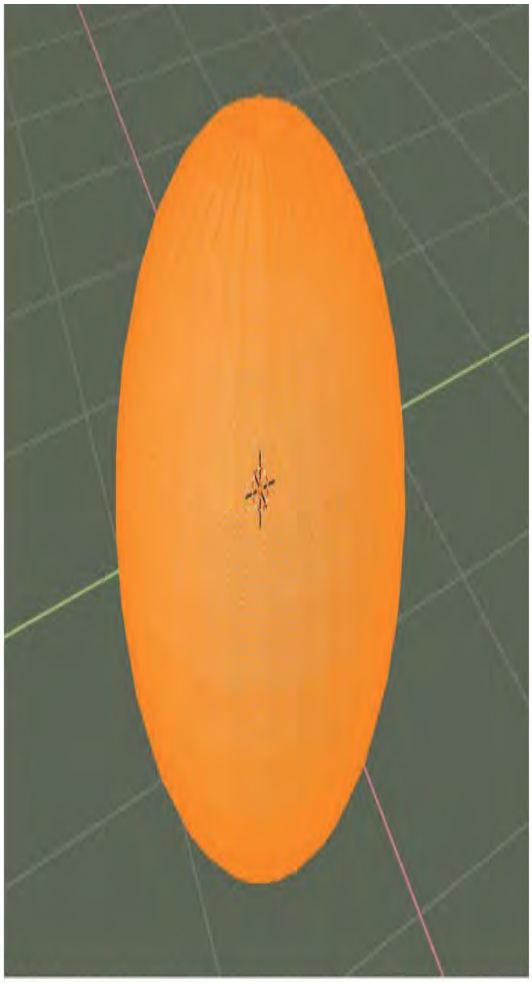
11.4 Sculpting

Sculpting, in Blender, allows you to add detail to the surface of a model by manipulating the Mesh Vertices. The process produces results similar to kneading a piece of clay. Vertices are pulled or pushed or added, by using of Tools which deform the mesh surface.

Sculpting is performed after you have created a model. To demonstrate the basics of the process a UV Sphere will be employed. The UV Sphere has a reasonable number of Vertices forming its surface but for Sculpting to be effective a high vertex count is required. Replace the default Cube Object with a UV Sphere in the 3D Viewport Editor. In the demonstration the whole surface of the Sphere will be used but in reality you would Subdivide the surface of a model in the area where detail is to be added.

With the UV Sphere selected in Object Mode, Tab to Edit Mode and with all Vertices selected, Subdivide with Number of Cuts 10 (Chapter 5 – 5.5) (Figure 11.20).

Last Operator Panel



UV Sphere – Edit Mode – All Vertices Selected

Figure 11.20

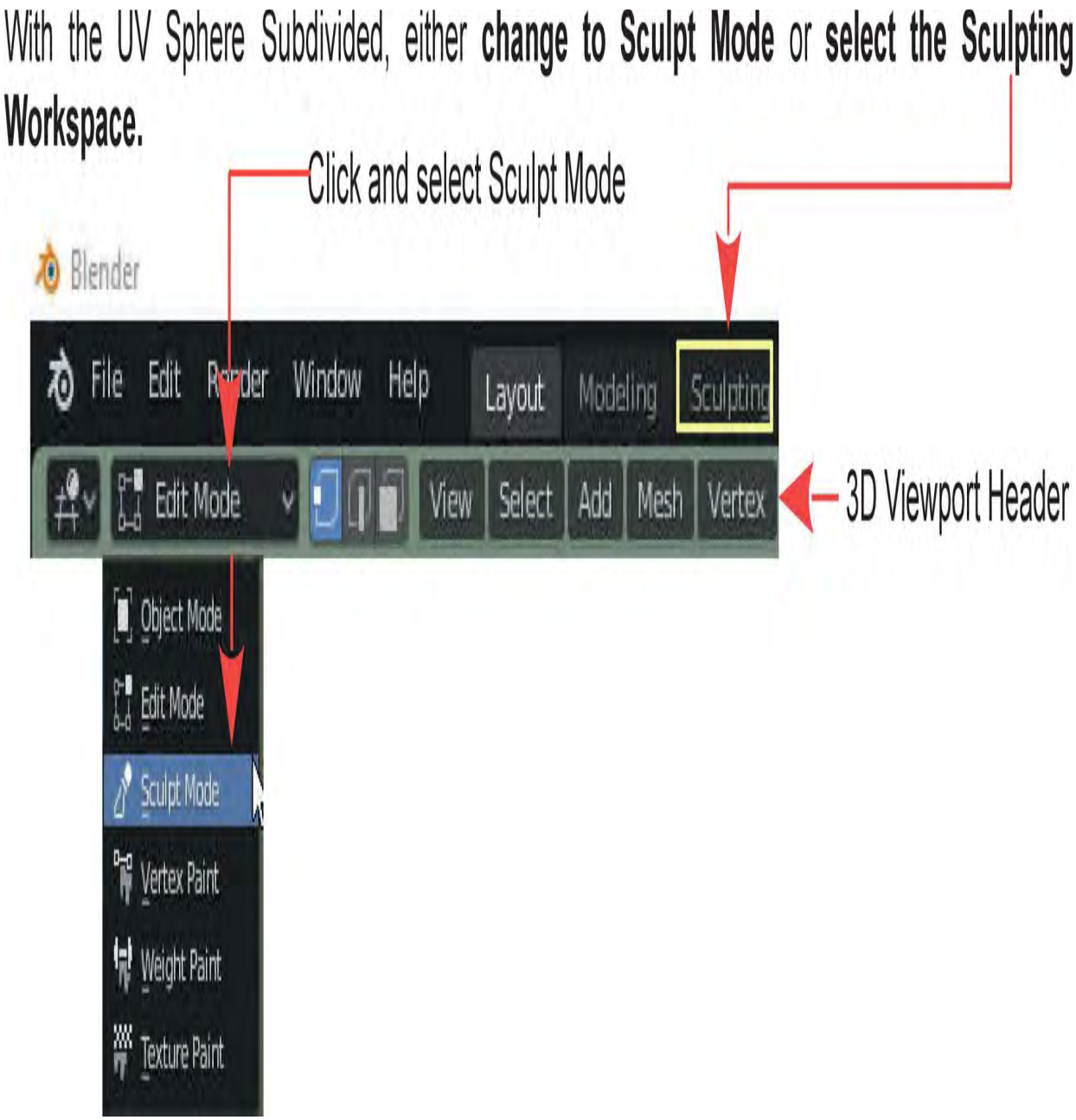


Figure 11.21

Note: Sculpt Mode is not available from the 3D Viewport Header unless an Object is present in the 3D Viewport Editor. The Sculpting Workspace will open, without an Object present and will allow you to Add Objects but again, Sculpt Mode is not available until an Object is present.

The first thing you will notice when changing to Sculpt Mode is the change in the 3D Viewport Editor Header and the arrangement of Tools down the LHS of the Screen. The second observation is, the subdivided UV Sphere is displayed as it would be in Object Mode with with Flat Shading. Changing back to Object Mode and selecting Shading - Smooth has no effect.

A third significant change is; the 3D Viewport Editor Cursor has a blue circle attached **when you mouse over the Sphere.** The circle is called the **Brush** (Figure 11.22).

Note: The color of the Brush Circle changes depending on the Tool selected.

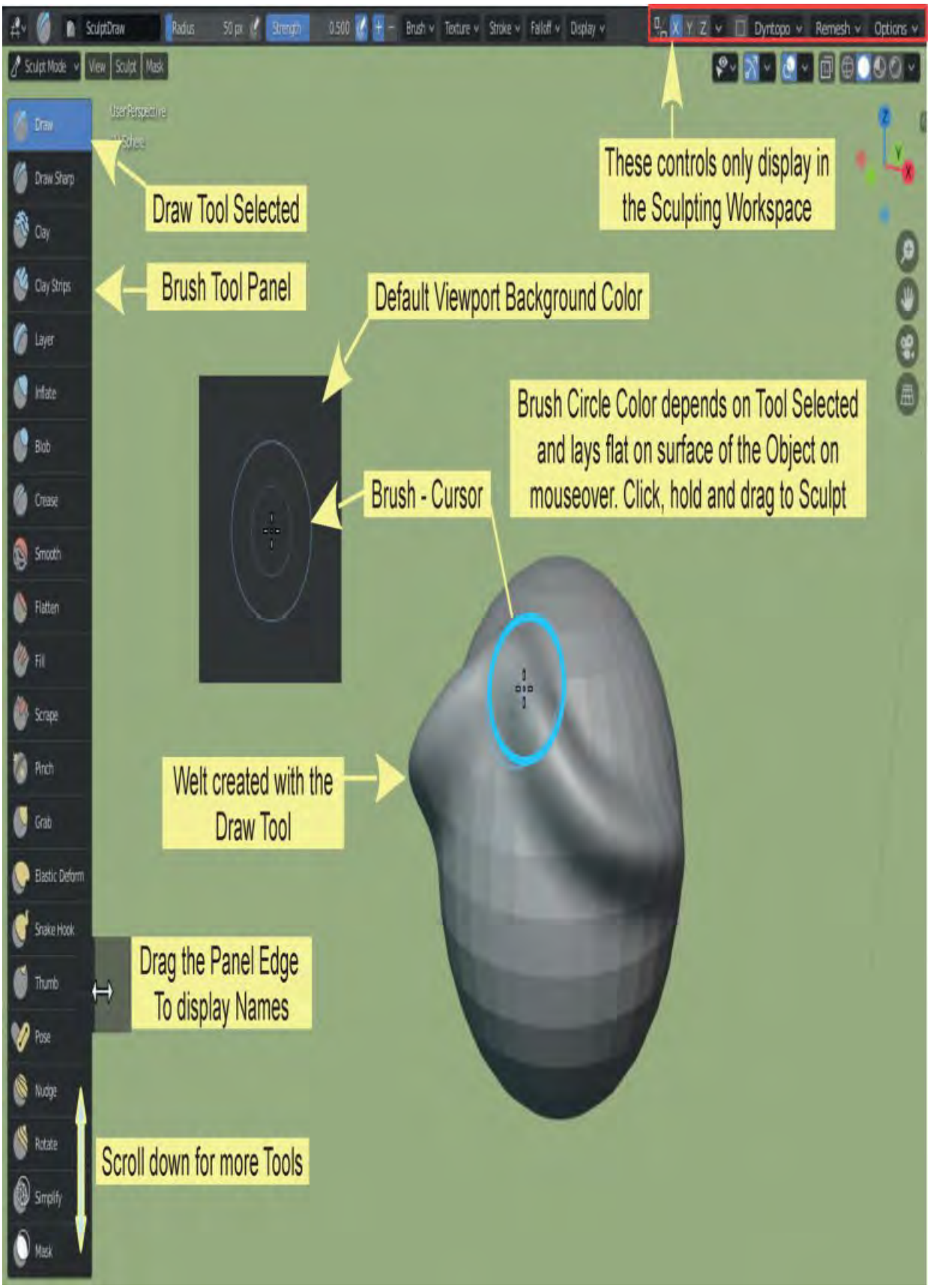

These controls only display in the Sculpting Workspace
Draw Tool Selected
Brush Tool Panel
Default Viewport Background Color
Brush Circle Color depends on Tool Selected and lays flat on surface of the Object on mouseover. Click, hold and drag to Sculpt
Brush - Cursor
Welt created with the Draw Tool
Drag the Panel Edge To display Names
Scroll down for more Tools

Figure 11.22

Note: This display shows the 3D Viewport Editor in Sculpt Mode when the **Sculpting Workspace** is selected in the Screen Header.

For example, with the Draw Tool selected (highlighted blue) click, hold and drag the mouse cursor (blue circle) over the surface of the UV Sphere. With the default settings Vertices forming the surface will be raised forming a welt. Note the deformation may be mirrored about a centerline with Mesh Symmetry set in the Header.

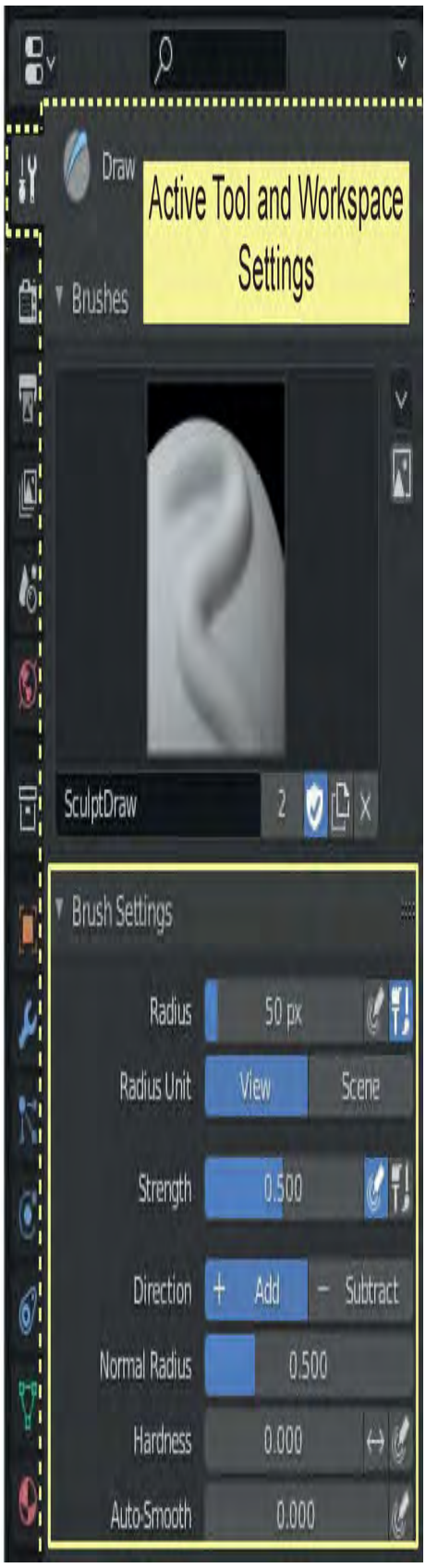
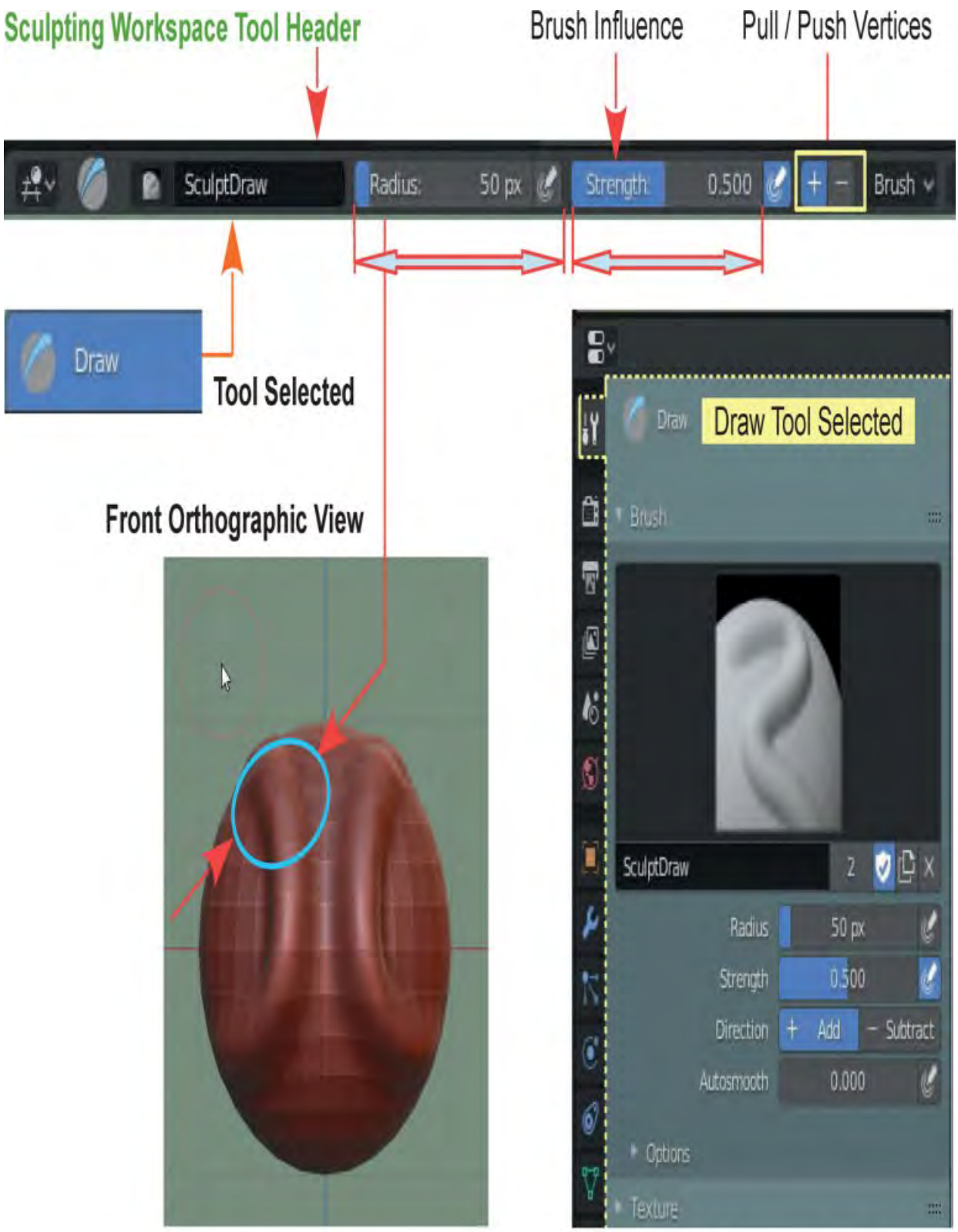


Figure 11.23

To Sculpt you select a Tool, click, hold and drag over the surface of the selected Object. In Figure 11.22 the Draw Tool has been selected (blue icon in the Tool Panel = blue Brush Circle). The Sculpt effect is dependent on the Strength setting in the Header and the Brush Radius.

For detailed Sculpting you would select the Sculpting Workspace. Detailed controls for each Brush Type are found in the Properties Editor, **Active Tool and Workspace settings** (Figure 11.23).



Controls duplicated in the Properties Editor

Figure 11.24

Note: In this instance the affect of the Brush is cumulative. The Brush has been dragged over the same Vertices three times **with the Strength value = 0.500**

Controls duplicated in the Properties Editor

11.5 Sculpting Demonstration (Basic)

The Sculpt demonstration will be limited to the Draw Tool. Each Tool affects the Mesh Surface in a different way and the results vary depending on what settings are applied to the individual tools. As with any graphics drawing application proficiency is obtained by experimentation and practise.

It is worth noting that instead of the Mouse a Graphics Drawing Tablet with a Stylus may be substituted.

To see how the Draw Tools operates, select a Draw tool in the Tool Panel. The buttons in the Tool Header will control the properties of the Draw Tool. Set the Radius slider to approximately 30 px (pixels) and have the plus + setting for Pull engaged. Change the Strength value to 1.000 (drag the slider). The effect of the Tool will have a more pronounced effect with a higher value.

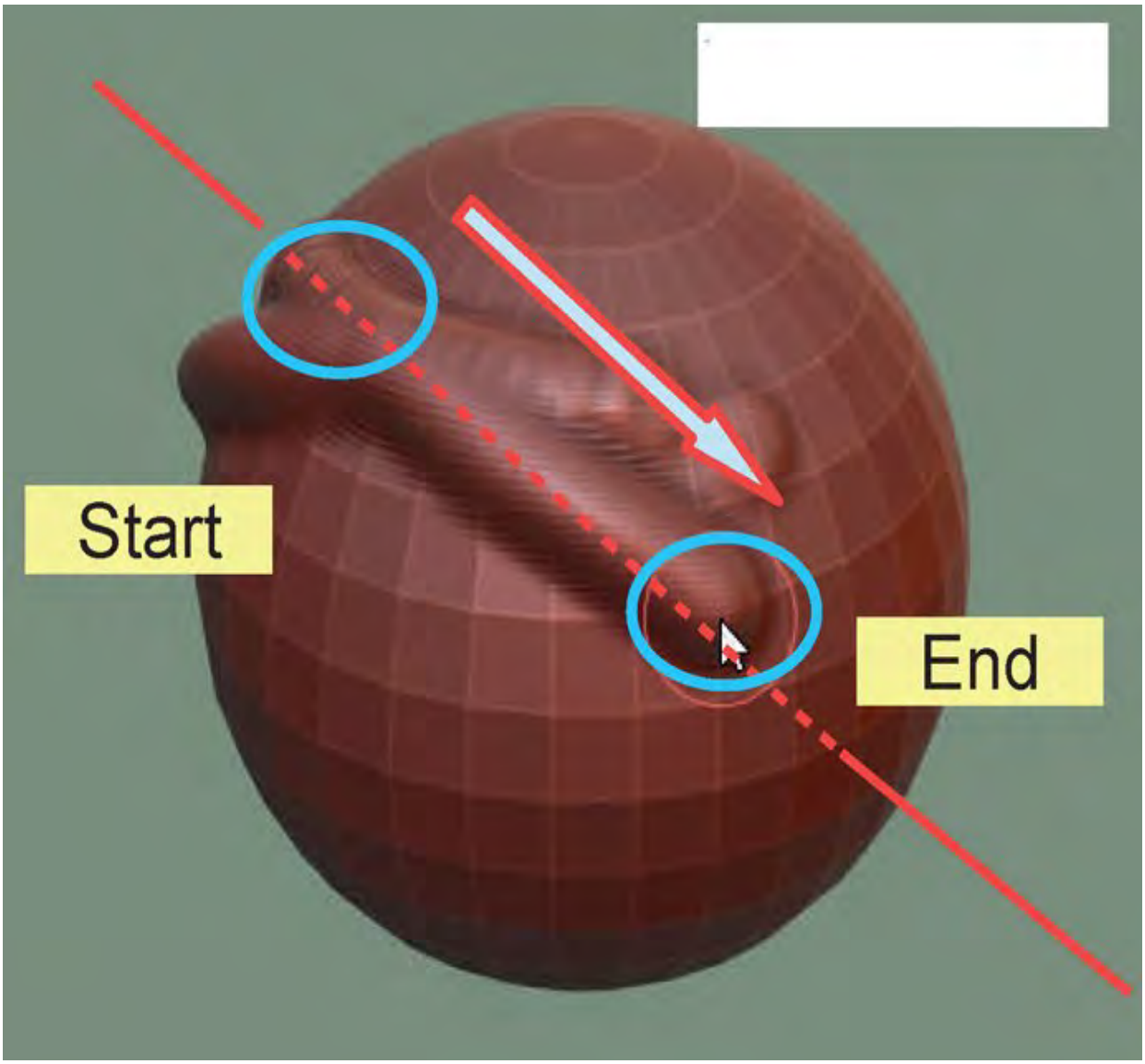


Figure 11.25

Position the Brush in the 3D Viewport Editor as shown in Figure 11.25, click and hold LMB and drag the Brush over the surface of the UV Sphere (generally along the red X Axis). At the end of the stroke release the mouse button.

As the Brush moves over the surface a welt appears as the Vertices are pulled away from the surface. Repeatedly dragging the Brush over the top of the welt further increases the deformation (Figure 11.26).

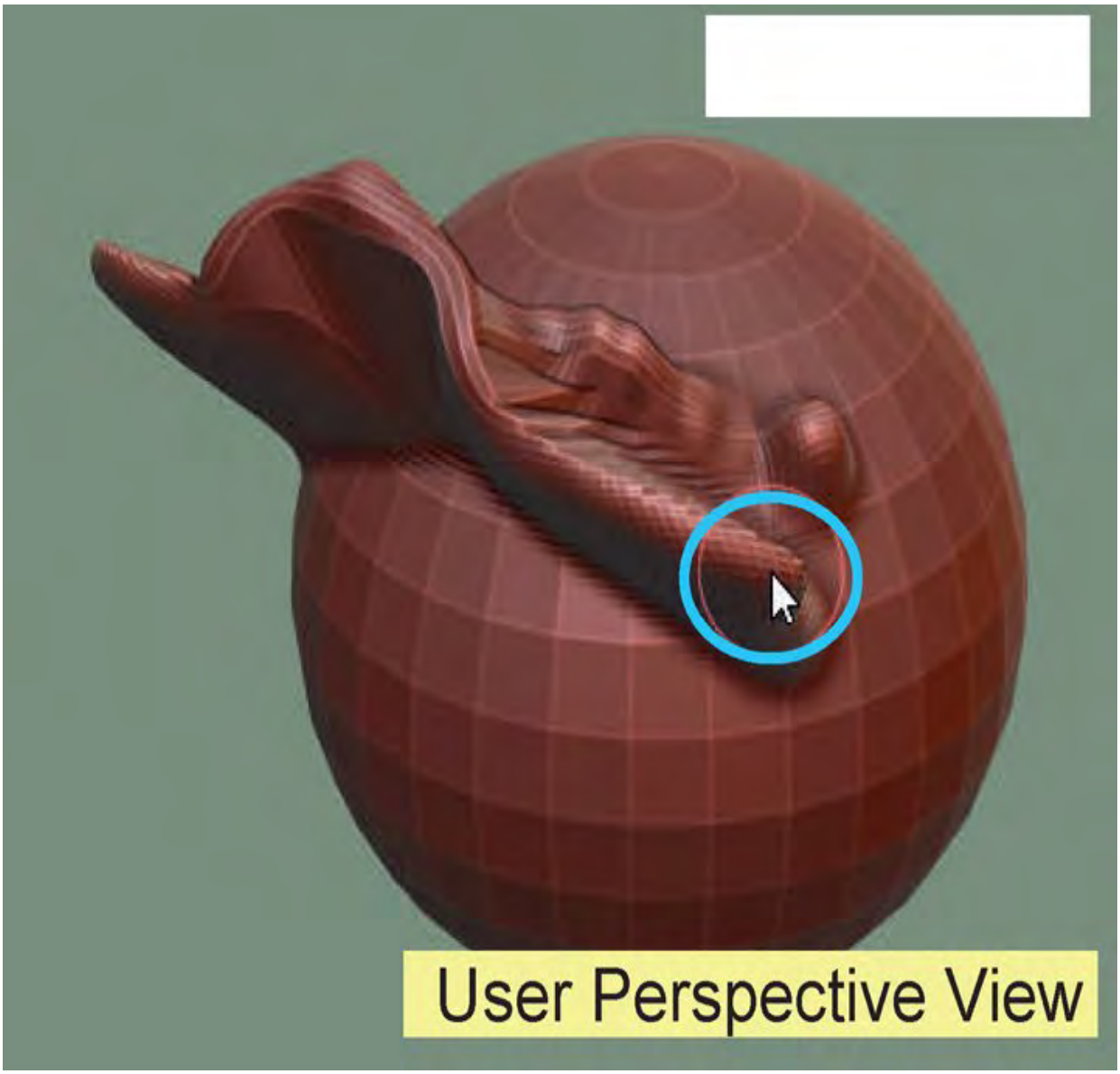


Figure 11.26

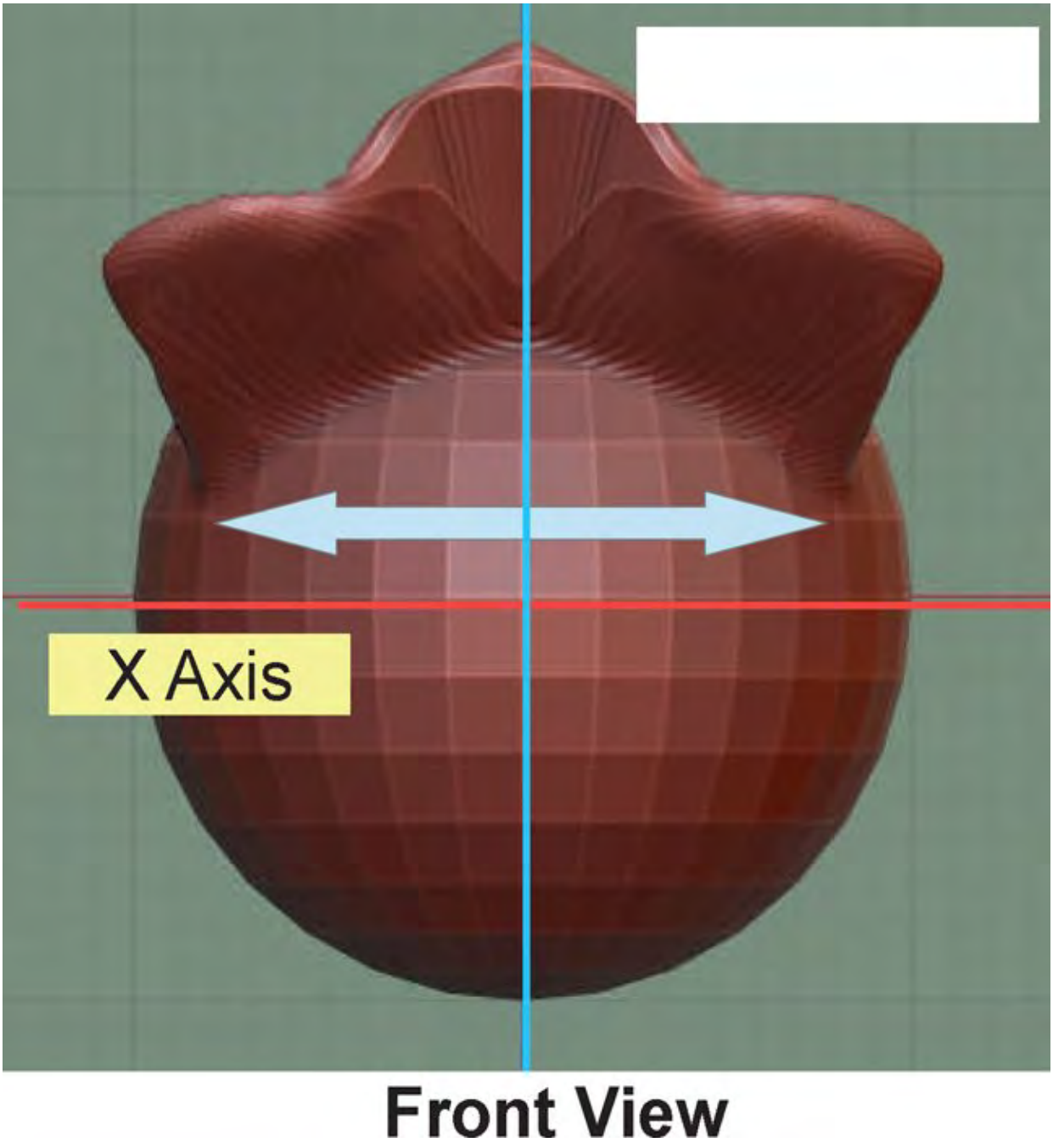


Figure 11.27

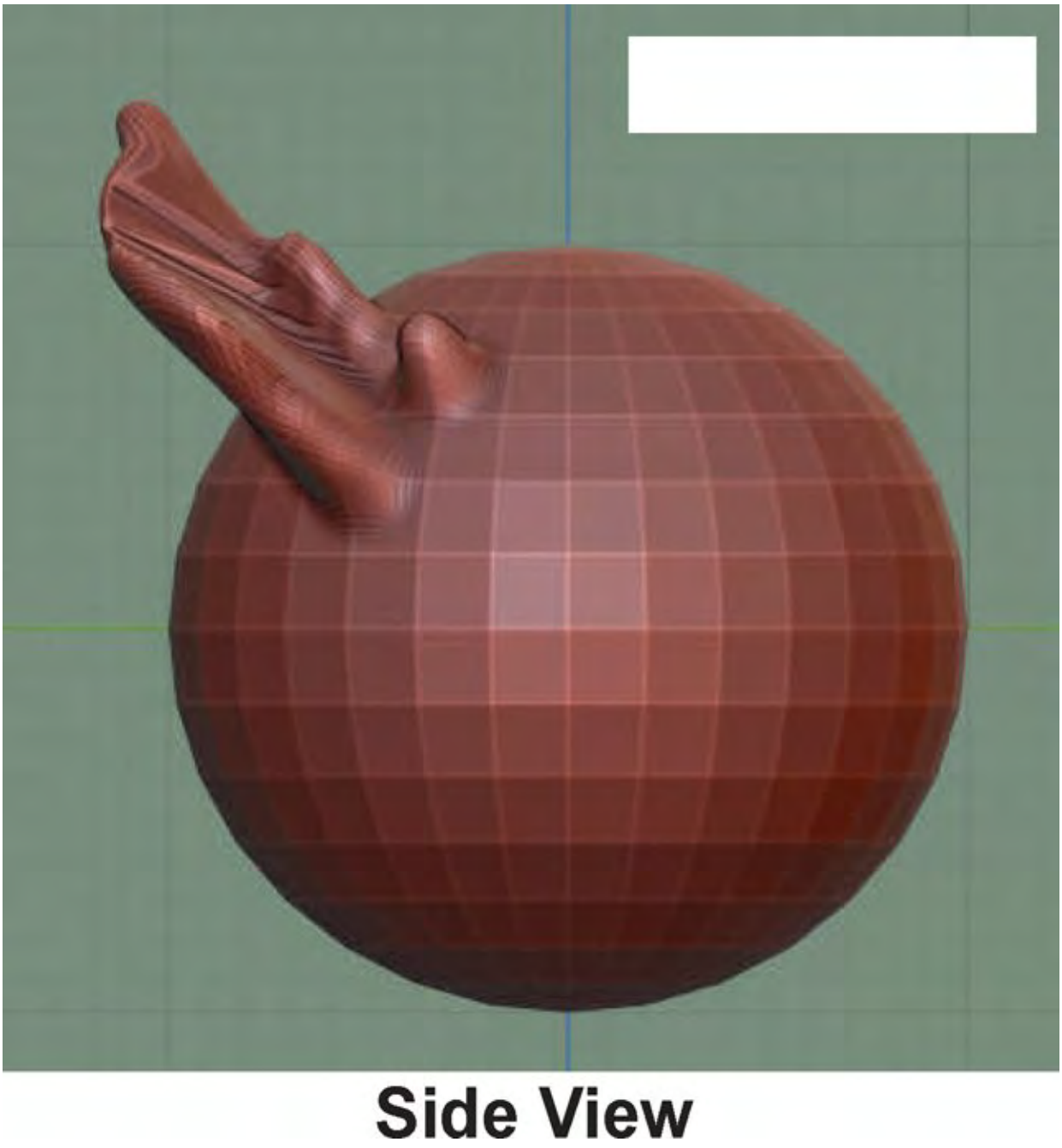


Figure 11.28

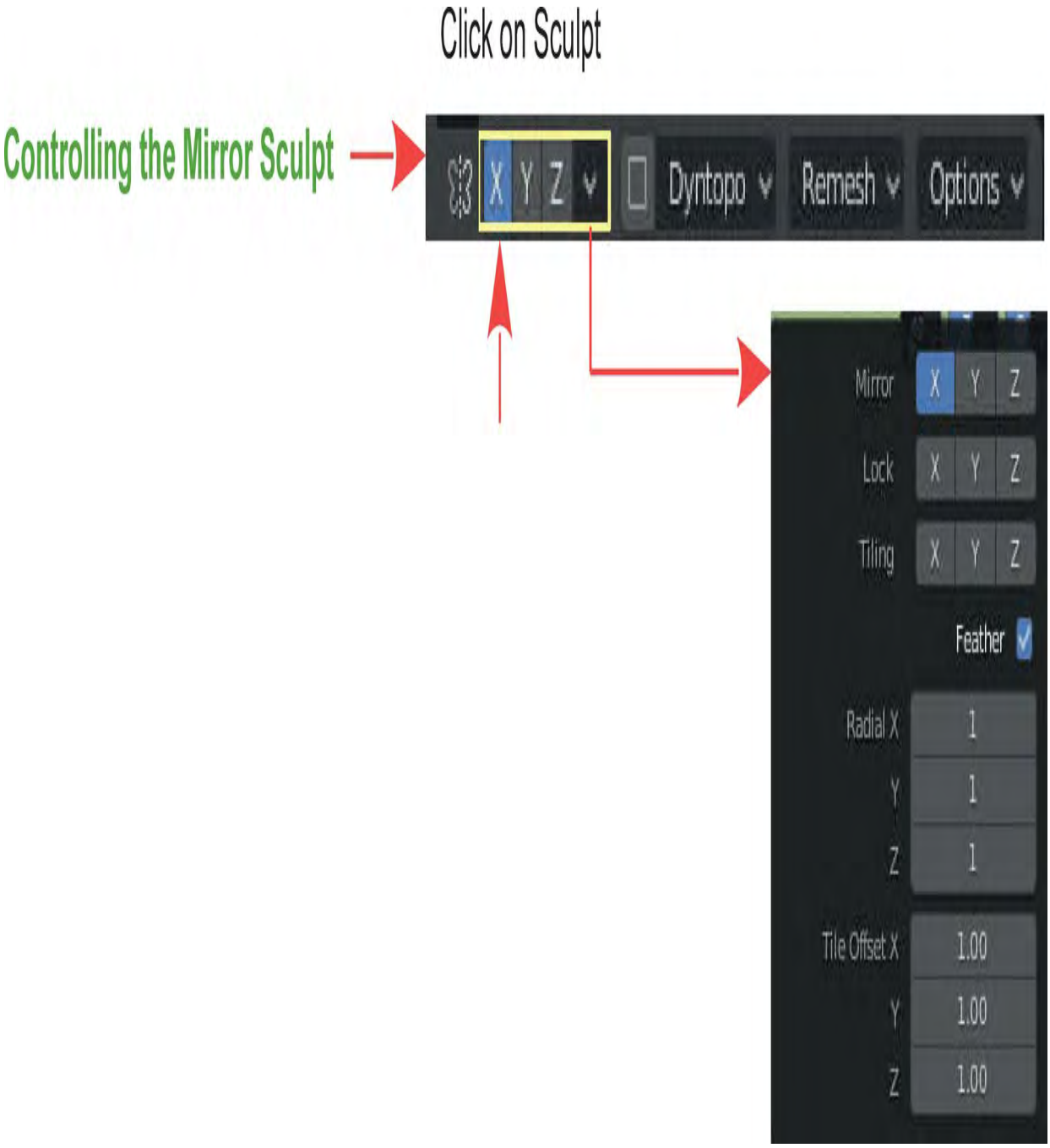


Figure 11.29

3D Viewport Editor Header Sculpting Workspace RHS

In the Front View you see that the deformation of the surface has been mirrored along the X Axis, that is mirrored either side of the Z Axis. To turn the mirror effect off, in the 3D Viewport Editor Header (in Sculpt Mode)

click on **X** to toggle **X Axis Mirror.** You may also elect to mirror the Sculpting about the Y or Z Axis.

With Mirror toggled off you can stroke over the surface in any direction modeling freehand.

At this stage clarification in respect to the relationship between the options available in the Headers is required. Figure 11.30 shows the options available in the Headers.

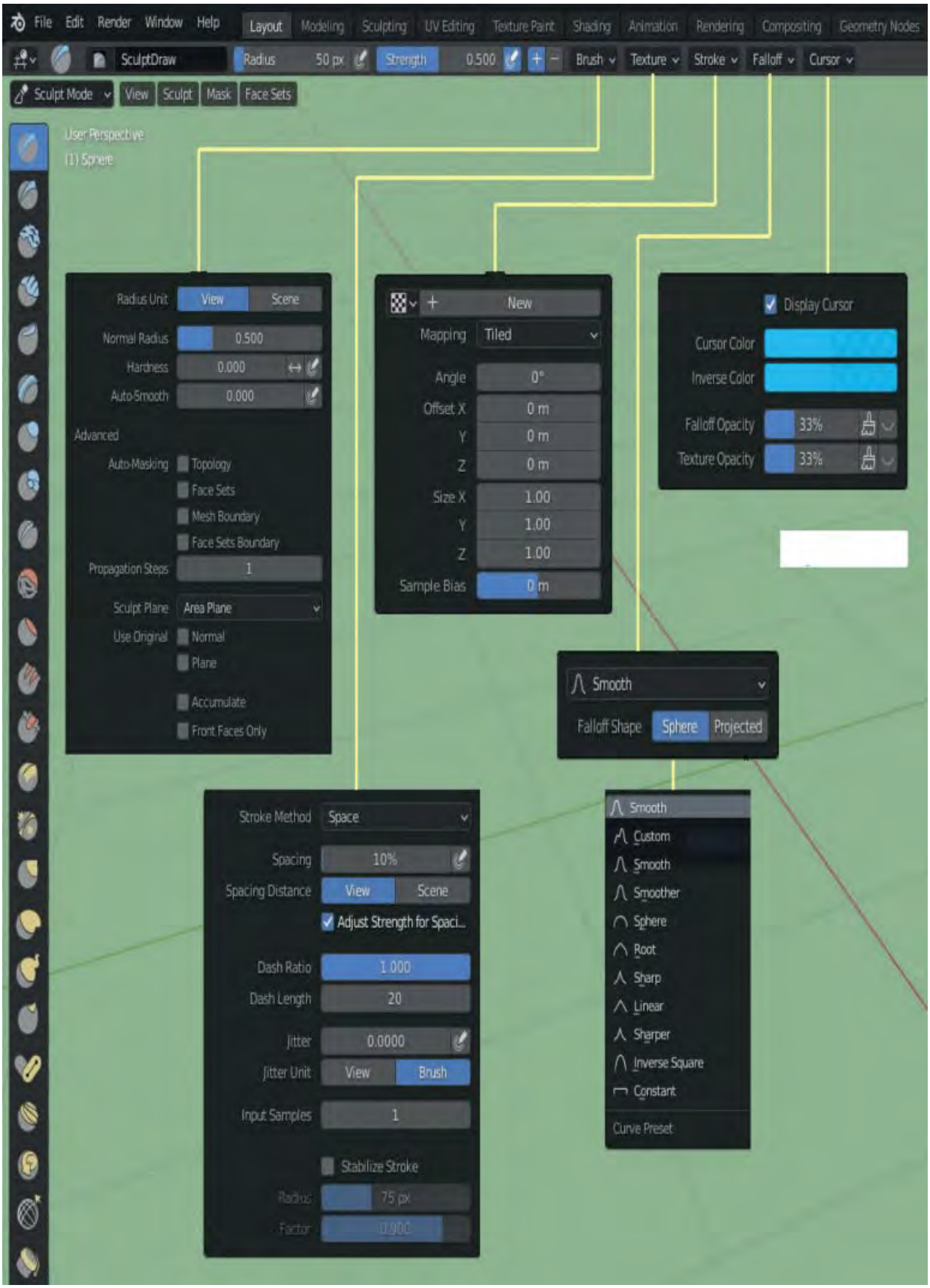

File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Geometry Nodes
SculptDraw
Radius 50 px
Strength 0.500
Brush Texture Stroke Falloff Cursor
Sculpt Mode View Sculpt Mask Face Sets
User Perspective
(1) Sphere
Radius Unit View Scene
Normal Radius 0.500
Hardness 0.000
Auto-Smooth 0.000
Advanced
Auto-Masking Topology
Face Sets
Mesh Boundary
Face Sets Boundary
Propagation Steps 1
Sculpt Plane Area Plane
Use Original Normal
Plane
Accumulate
Front Faces Only
New
Mapping Tiled
Angle 0°
Offset X 0 m
Y 0 m
Z 0 m
Size X 1.00
Y 1.00
Z 1.00
Sample Bias 0 m
Display Cursor
Cursor Color
Inverse Color
Falloff Opacity 33%
Texture Opacity 33%
Smooth
Falloff Shape Sphere Projected
Stroke Method Space
Spacing 10%
Spacing Distance View Scene
Adjust Strength for Spaci...
Dash Ratio 1.000
Dash Length 20
Jitter 0.0000
Jitter Unit View Brush
Input Samples 1
Stabilize Stroke
Radius 75 px
Factor 0.900
Smooth
Custom
Smooth
Smoother
Sphere
Root
Sharp
Linear
Sharper
Inverse Square
Constant
Curve Preset

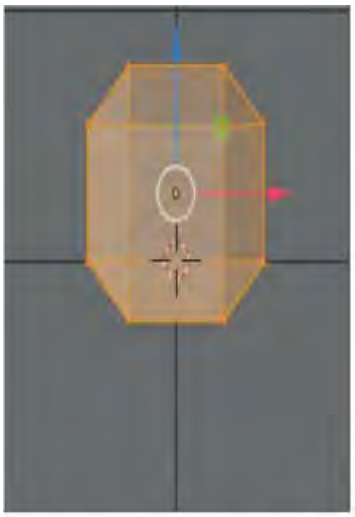
Figure 11.30

Controls in the Header alter depending on the particular Brush selected from the Brush Panel. In the diagram the Draw Brush is selected (highlighted blue). You see the icon representing Draw at the LHS of the Header.

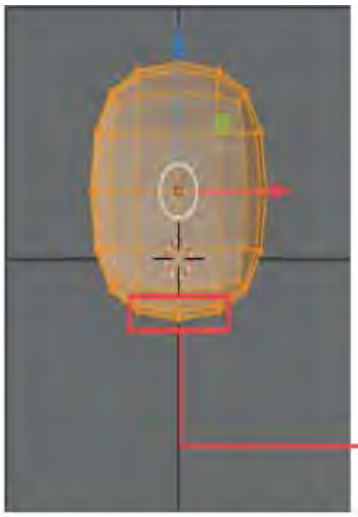
The Headers provide a variety of options for selecting how the Brush affects the mesh surface. Many of the selection options are duplicated. Knowing what Tools are available and where to access them will allow you to experiment and practise sculpting.

11.6 Creating a Humanoid Figure

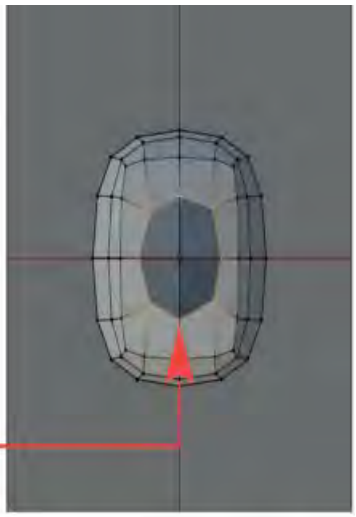
At some stage you will want to make a character figure for animation. As with the Aircraft Modeling Exercise this starts with one of Blender's primitives (a Cube) and by a process of selecting Vertices, Extrusion, Scaling, Manipulation and the application of Modifiers a simple figure may be generated. The following is intended to demonstrate the technique and not to produce a refined result. Modeling requires time and patience and plenty of practise. Add a **Bevel Modifier** to the default Cube with Width = 0.401. Apply the Modifier and place the Cube in **Edit Mode — Front Orthographic View. Remember: Modifiers can only be Applied in Object Mode.**



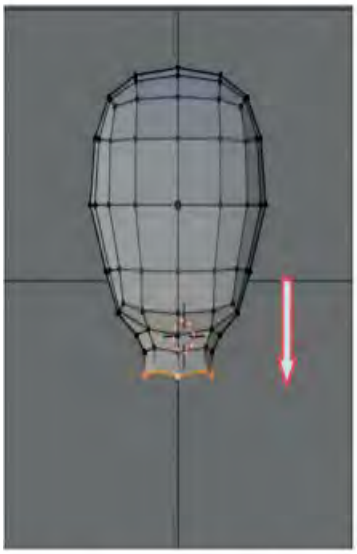
Cube with Bevel Modifier applied. Width = 0.401



Apply a Subdivision Surface Modifier to add Vertices.

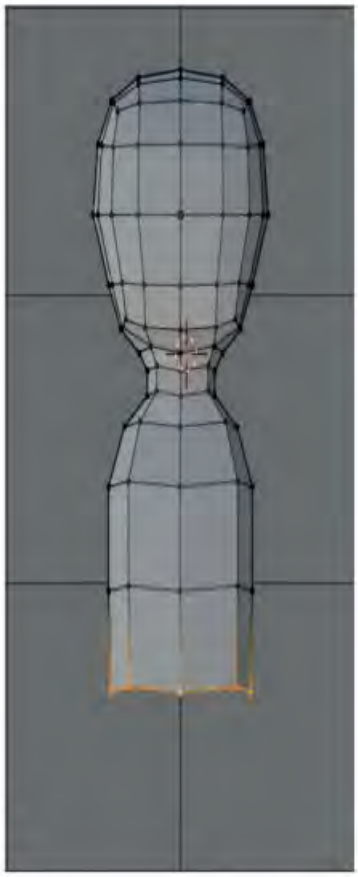


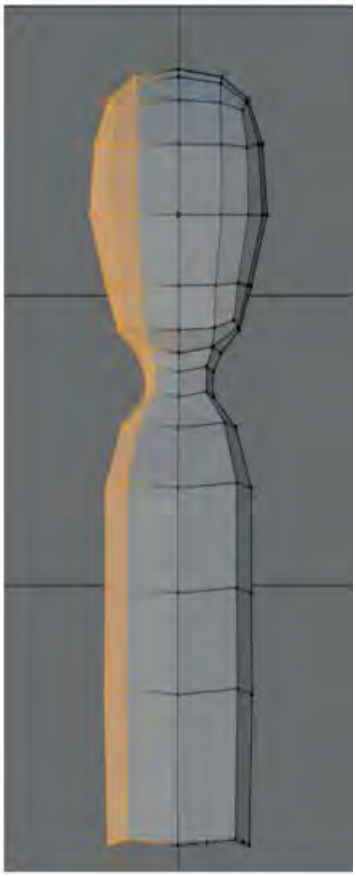
Remove the center Vertex and reshape perimeter of the hole on the base.



Select perimeter Vertices and Extrude down to form the neck.

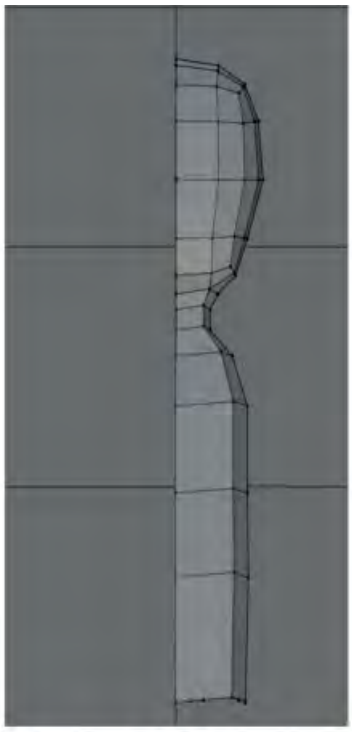
Continue extrusions and Scale out forming the body.





With the body formed select one half of the Vertices and delete (Front View).

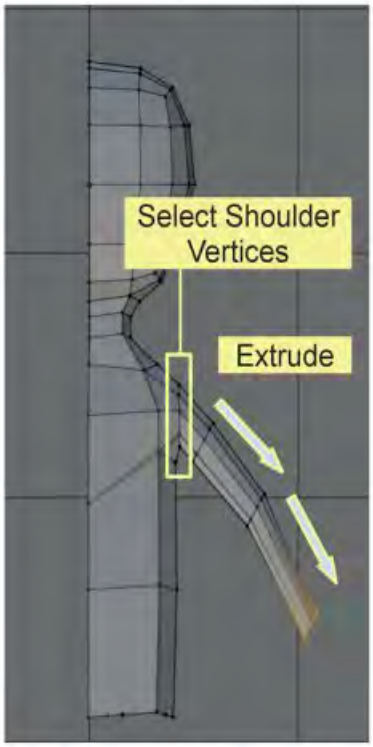
Figure 11.31
Cube with Bevel Modifier applied. Width = 0.401
Apply a Subdivision Surface Modifier to add Vertices.
Remove the center Vertex and reshape perimeter of the hole on the base.
Select perimeter Vertices and Extrude down to form the neck.
Continue extrusions and Scale out forming the body.
With the body formed select one half of the Vertices and delete (Front View).



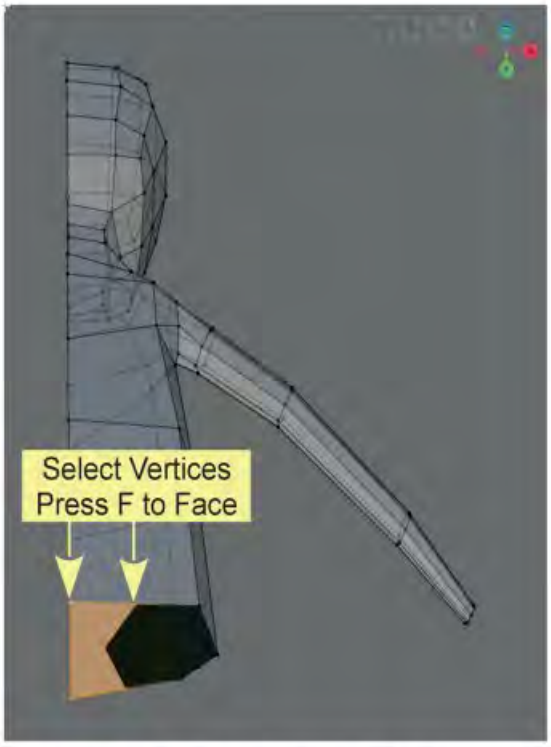
Edit Mode

Object Mode

Add a Mirror Modifier in Object Mode (Do **NOT** Apply).



In Edit Mode, select Vertices in the shoulder. Delete one Vertex to create a hole. Extrude and Scale the arm.



At the base, Subdivide to create Vertices and Face the center portion leaving a hole for the leg.

Figure 11.32

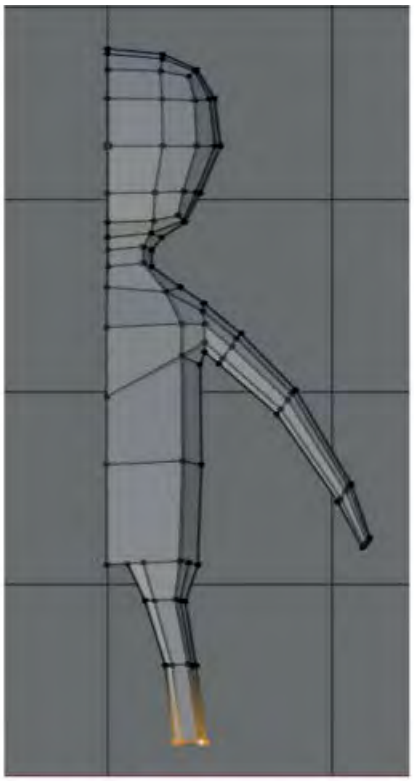
Edit Mode

Object Mode

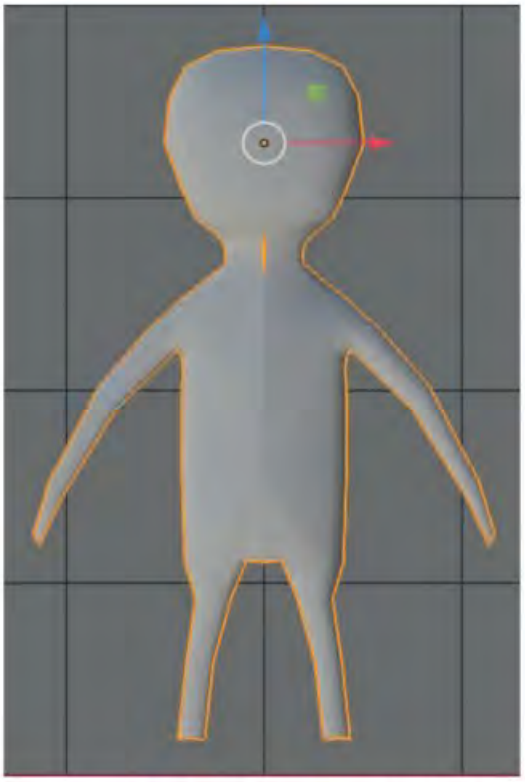
Add a Mirror Modifier in Object Mode (Do **NOT** Apply).

In Edit Mode, select Vertices in the shoulder. Delete one Vertex to create a hole. Extrude and Scale the arm.

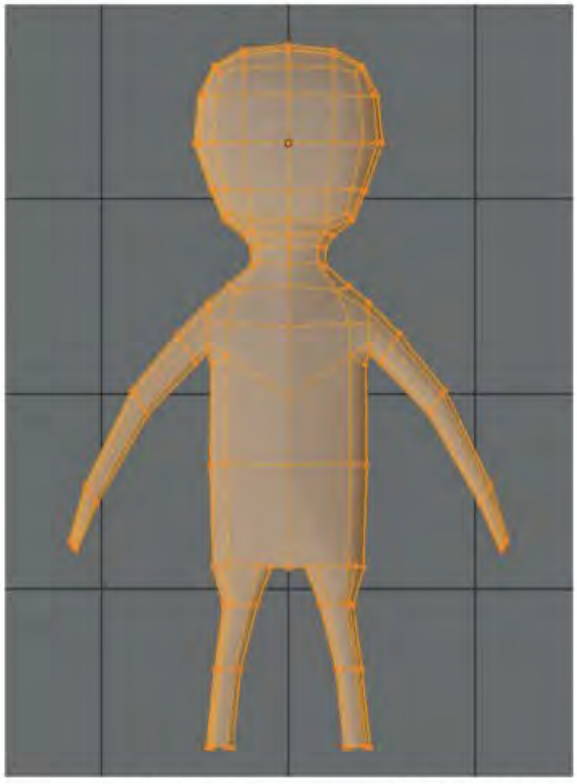
At the base, Subdivide to create Vertices and Face the center portion leaving a hole for the leg.



Extrude, Scale and position Vertices forming a leg and foot (not shown).



In Object Mode set Shading to Smooth and **Apply the Mirror Modifier**.



Edit Mode showing Vertices when the Mirror Modifier is Applied

Figure 11.33

Extrude, Scale and position Vertices forming a leg and foot (not shown).

In Object Mode set Shading to Smooth and **Apply the Mirror Modifier.**

Note: This is a very rough crude model shown for modeling technique only. **See Chapter 22**

Edit Mode showing Vertices when the Mirror Modifier is Applied

12 The Outliner and Collections

12.1 Collections in the Outliner Editor
12.2 View Options in the Outliner Editor

The Outliner Editor

The **Outliner Editor** provides a visual display in the form of a **File Tree** showing everything in the Scene in the **3D Viewport Editor.** Each Object is listed showing its relationship to other Objects such as whether it is joined to another Object or has a Child Parent Relationship or whether it has Material or Texture applied. This information is the **Object's Data** which may be displayed in different configurations depending on the function being performed.

Collections

The File Tree in the **Outliner Editor** may be customised allowing Objects in the 3D Viewport Editor to be arranged in groups called **Collections.** Collections may be added and named and arranged in a hierarchy much the same as the folders on an operating system. Objects in the 3D Viewport Editor may be selected and deselected in the Outliner Editor or hidden from view in the 3D Viewport Editor by turning a Collection's visibility off. This assists when working on Objects in a complicated Scene.

12.1 Collections in The Outliner Editor

The **Outliner Editor** (upper RHS of the Screen) provides a visual display, in the form of a file tree, of everything in your Scene and shows how the

different items are related (Figure 12.1).

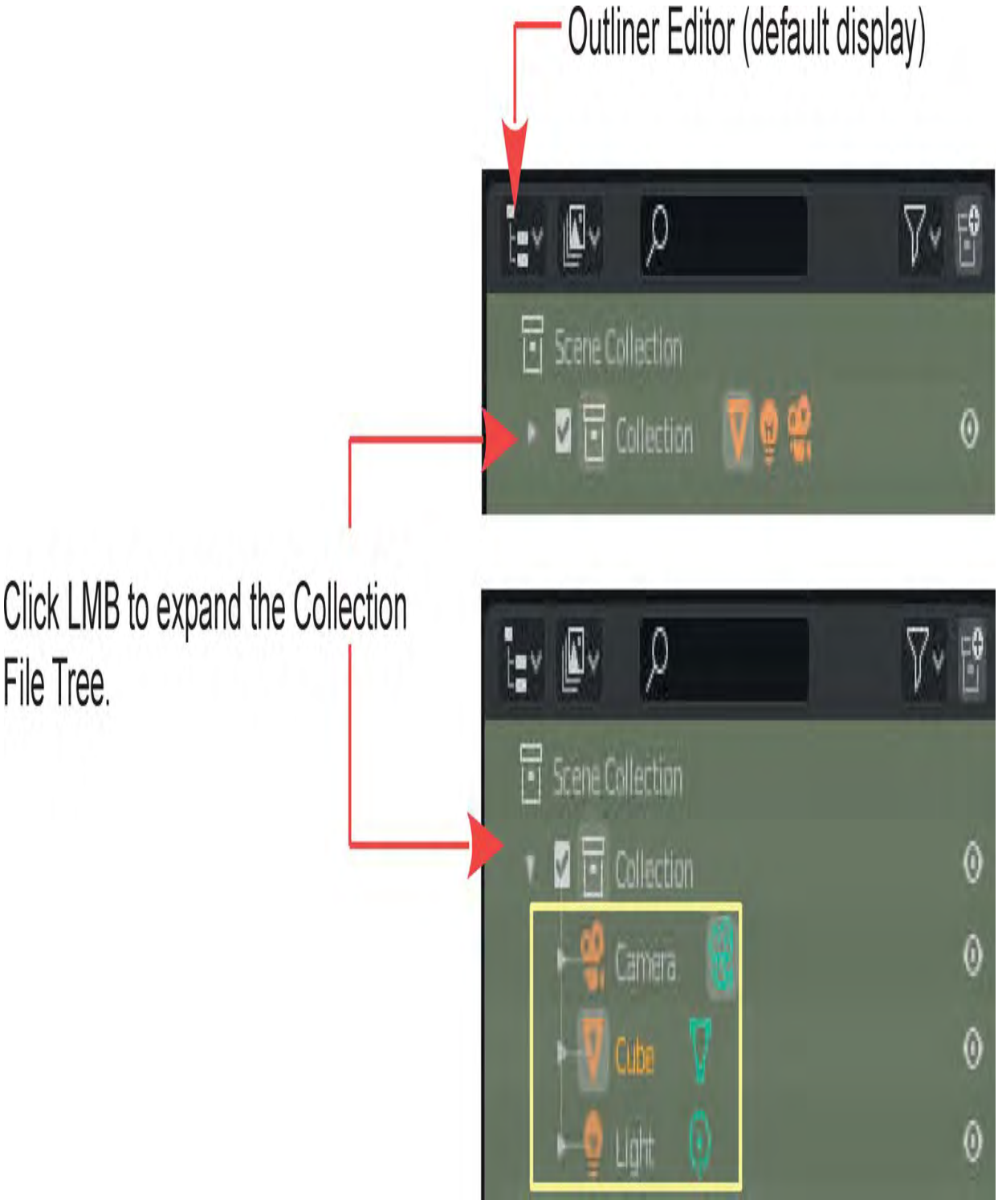


Figure 12.1

The Scene in the default 3D Viewport Editor contains a Camera, a Cube and a Light.

To understand how Collections operate work through the following procedure.

Start with the default Blender Screen showing the four default Editors. The **Outliner Editor** is displayed in the upper right hand corner of the Screen (Figure 12.1).

The **Outliner Editor** contains information about the current Scene in the default 3D Viewport Editor. If, under **Scene Collection,** the single entry **Collection** displays. Click on the **expansion icon** preceding Collection to expand the File Tree showing a list of Objects in the current Scene in the 3D View Editor.

Expansion Icon

The 3D Viewport Editor contains three Objects: a Cube, a Camera and a Light. The three Objects are grouped together and placed in the Collection named **Collection.**

Click on the expansion icon at the beginning of the line where you see **Collection** to show the Camera, Cube and Light listed. Note that in front of each Object there is also an expansion icon. You click on each icon to display data belonging to the Objects.

Clicking the expansion icon in front of Cube reveals; **Cube** (the data creating the display of the Cube in the 3D Viewport Editor), Clicking on the next expansion icon reveals **Material** (the data producing the gray color of the Cube). Each line represents a **Datablock** (block of data) producing the display in the 3D Viewport Editor.

Blue highlight indicates that the Cube Object is selected in the 3D Viewport.

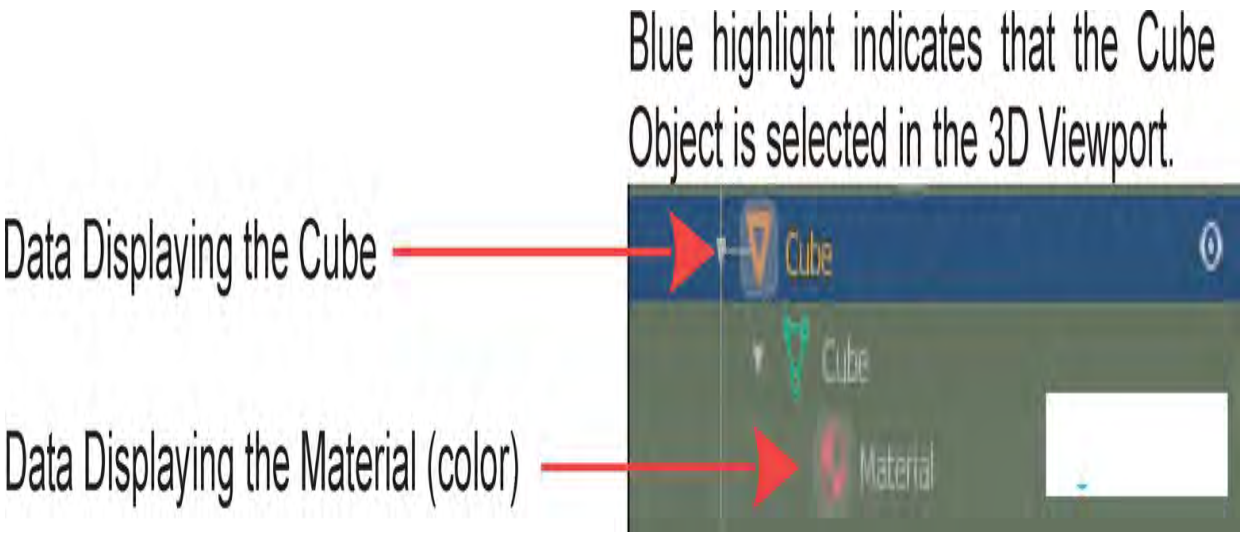


Figure 12.2

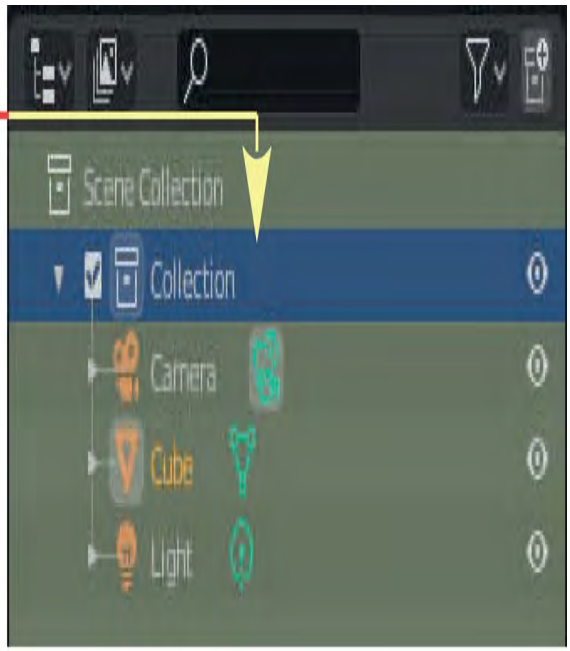
Adding Objects

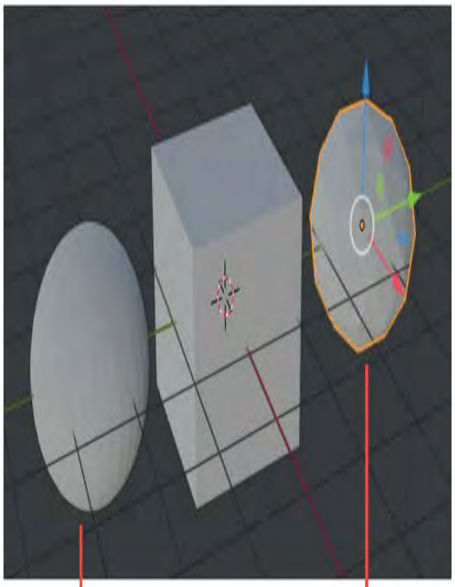
Clicking LMB on the Collection name selects the Collection (highlighted blue).

When you add a new Object to the Scene in the 3D Viewport it is automatically added to the Collection that is selected, in this case **Collection.**

Add a UV Sphere Object to the Scene followed by an Icosphere Object.

Clicking LMB on the Collection name selects the Collection (highlighted blue).





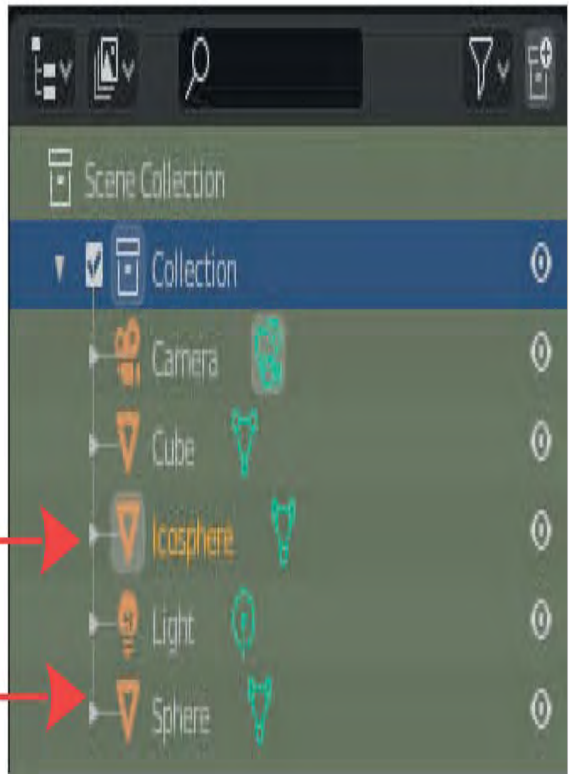


Figure 12.3

You immediately see **Icosphere** and **Sphere** added to **Collection.** Note that they are placed in the list alphabetically.

Adding Collections

As the Scene is developed, with Objects being added, it can be advantageous to create new Collections, grouping Objects together. You add (create) new Collections by clicking the **New Collection** button in the Outliner Editor Header.

Where Collections are added depends on the location you select before adding. If you select **Collection** in the Outliner Editor a new Collection will be added as a sub entry under Collection . The sub entry will be named Collection 1. If you select (highlight) **Scene Collection** before adding the new Collection it will be added as a sub Collection named Collection 2. Selecting **Collection 1** and adding a new Collection produces **Collection 1.1**

Note the Collection Names: New Collection 2 under Scene Collection and new Collection 1.1 under Collection 1.

Figure 12.4

You may rename Collection to something meaningful by double clicking on the Collection Name.

Deleting Collections

In the Outliner Editor you select the Collection name (highlights blue) and press delete to remove the Collection.

Be aware that deleting a Collection in the Outliner Editor does not delete Objects in that Collection. When you delete a Collection, Objects in the Collection are automatically transferred to the preceding Collection in the hierarchy.

Deleting Objects

You may select Objects in the Outliner Editor then RMB click and select delete in the menu to remove the Object from the 3D Viewport Editor. Deleting a Collection in the Outliner Editor transfers all Objects in the Collection to the preceding Collection in the hierarchy.

Hiding and Restricting Object Display and Selection

Objects in the 3D Viewport can be controlled in the Outliner Editor.

Click the Eye Icon to Hide the Object from View in the 3D View Editor

By default only the Eye Icon is displayed.

Additional controls may be added to the Outliner Editor from the **Filter Menu**.

All **Restriction Toggles** Activated

Filter Menu

The control options are applicable to individual Objects and Collections.

Click Sphere, hold and drag to Collection 2

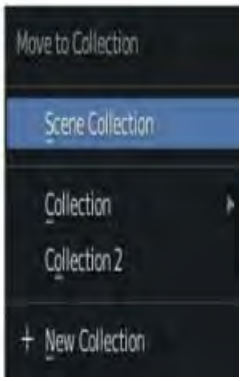


Figure 12.5

Moving Objects to Different Collections

Objects listed under a Collection in the Outliner Editor may be moved to a different Collection by simply clicking on the Object name (LMB), holding, dragging and releasing the mouse button with the Mouse Cursor positioned over the new Collection name. Alternatively press the **M Key** to display the **Move menu. Collections may be repositioned in a similar manner.**

Collection Properties

Collections are used to logically organize your scene, or to facilitate one-step appending or linking between files or across scenes.

A new addition to the Properties Editor is Collection Properties.

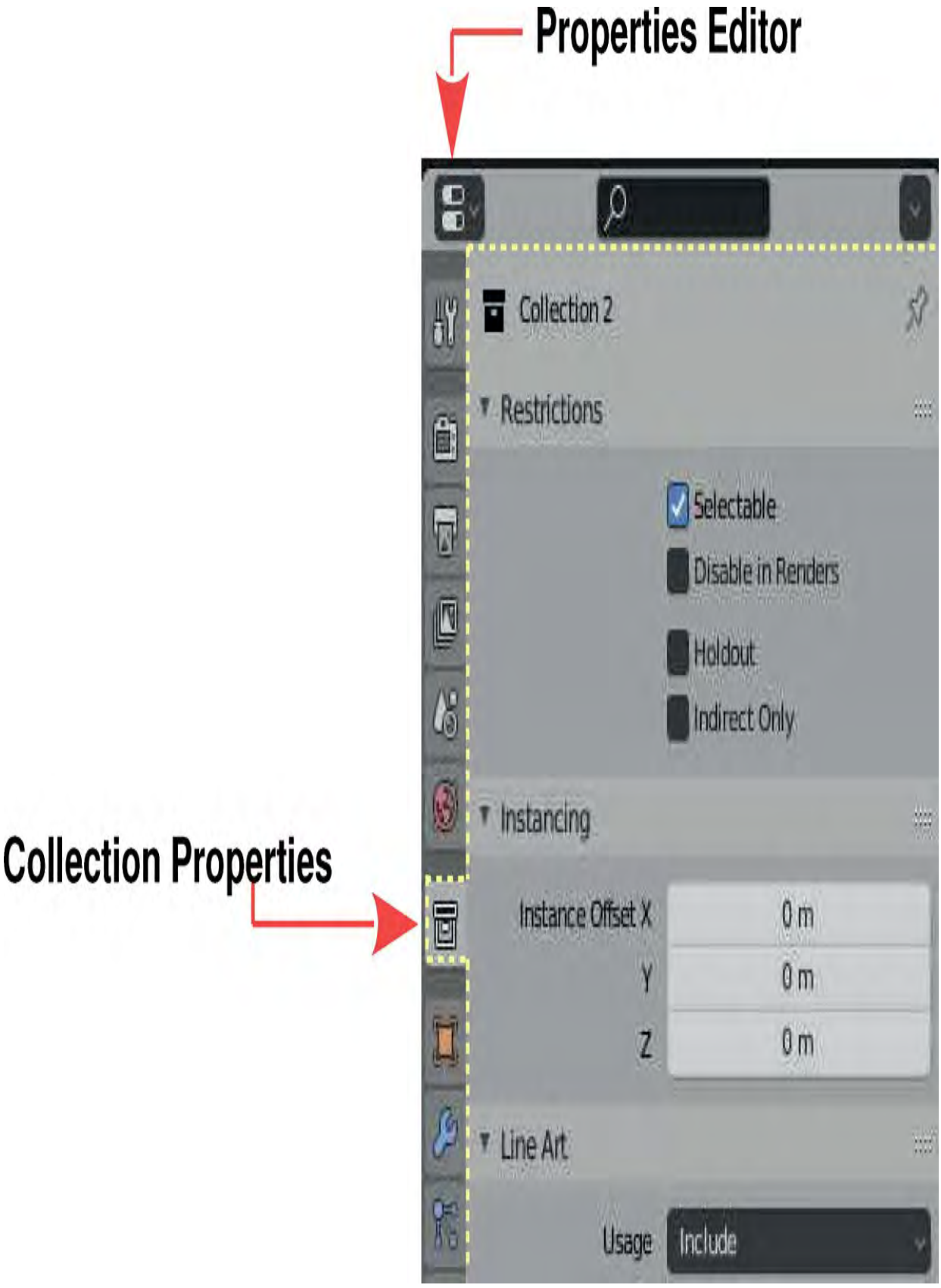

Properties Editor
Collection Properties
Collection 2
Restrictions
Selectable
Disable in Renders
Holdout
Indirect Only
Instancing
Instance Offset X
0 m
Y
0 m
Z
0 m
Line Art
Usage
Include

Figure 12.6

Restrictions

Selectable: Toggles selection of Objects in the Collection On/ Off.

Disable in Render: When selected, Objects in the Collection do not Render.

Holdout: Objects in the Collection generate a Holdout/Mask in the active View Layer.

Indirect Only: Objects only contribute to the Image through shadows and reflections.

Instancing

Applies a spatal offset from the original Object's Origin.

12.2 View Options in the Outliner Editor

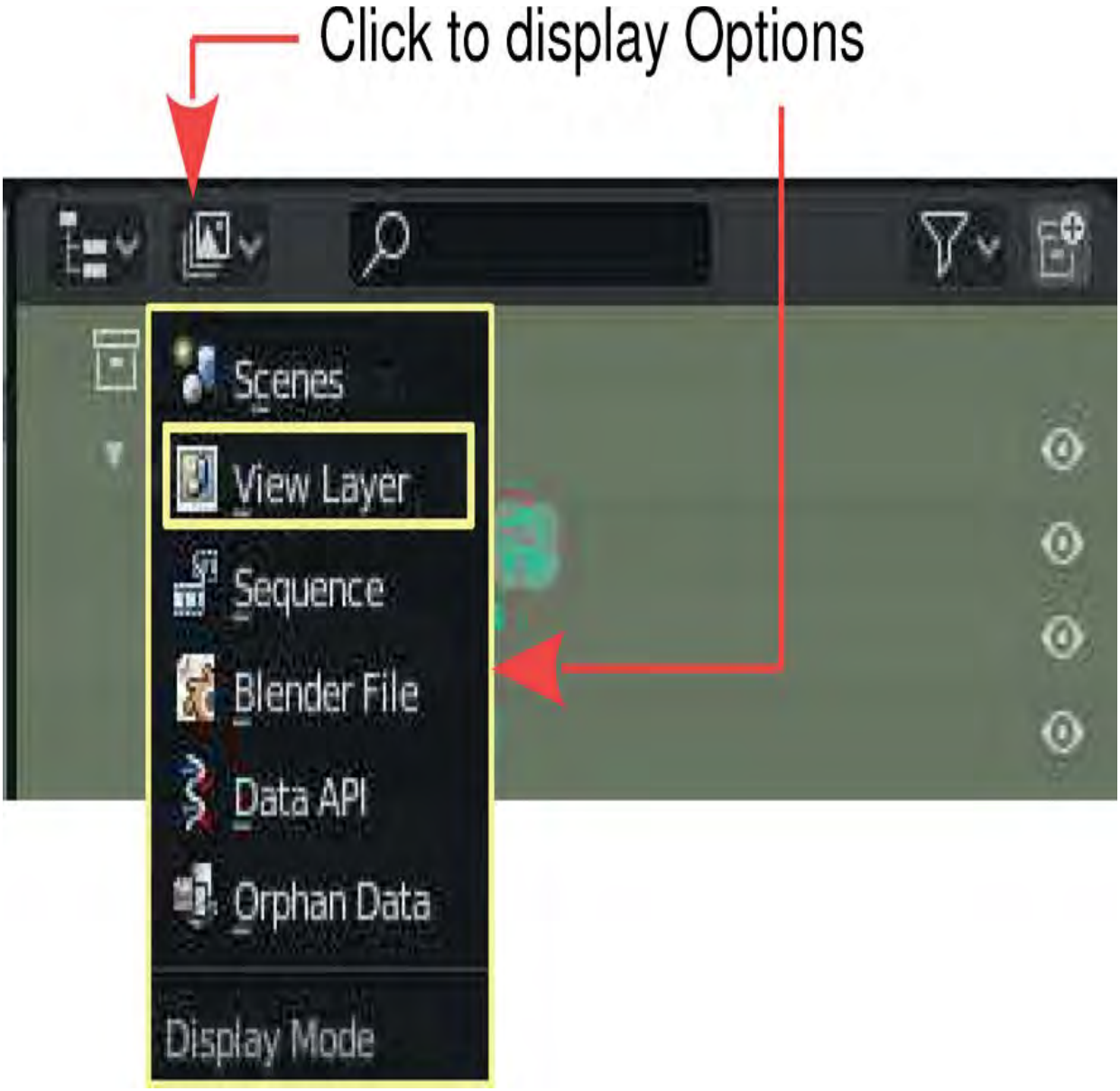


Figure 12.7

View Layer is the default display in the Outliner Editor which includes Collections.

Alternative options are accessed in the Editor Header. The alternatives display information about the Scene in the 3D Viewport Editor and the current Blender File.

As Scenes are developed the data contained in a Blender file increases accordingly. The display options in the Outliner Editor provide a record of

the data and allow it to be organised.

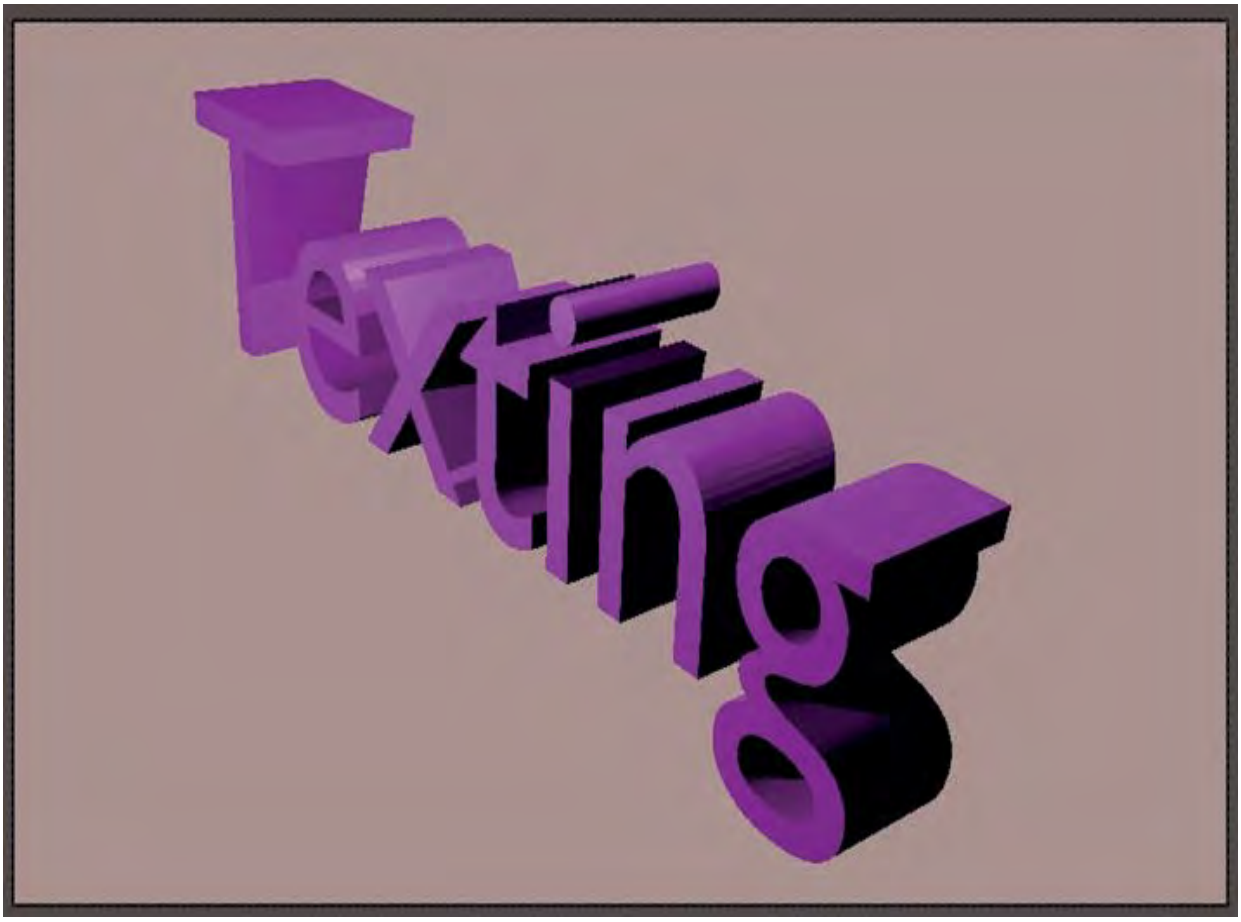
13 3D Text

13.1 Creating 3D Text in Blender
13.2 The Object data Button “F”
13.3 Fonts
13.4 Creating Text on a Curve
13.5 Converting Text to a Mesh Object
13.6 Converting Text to a Curve
13.7 Entering External Font

Introduction to 3D Text

3D text can be a very important element to add to a Scene. Think of all the television advertisements that contain text and how it is animated. There are two ways of adding text to a Scene in Blender. One way is to use the built-in text generator and the other is to use an external program.

Text made in Blender can be easily edited in the **Properties Editor.** Text made in an external online 3D text generating programs may give you additional options and different fonts.



13.1 Creating 3D Text in Blender

To create Text in Blender, place the Scene in **Top Orthographic View** (Num Pad 7). Delete the Cube Object. Select the **Cursor Tool** in the Tool Panel and locate the **Editor Cursor** at the point in the Scene where you want your text to go. Press **Shift + A key** and select **Add - Text** (Figure 13.1). The word **Text** displays in the 3D Viewport Editor in **Object mode. Note:** Text only displays in Top Orthographic View.

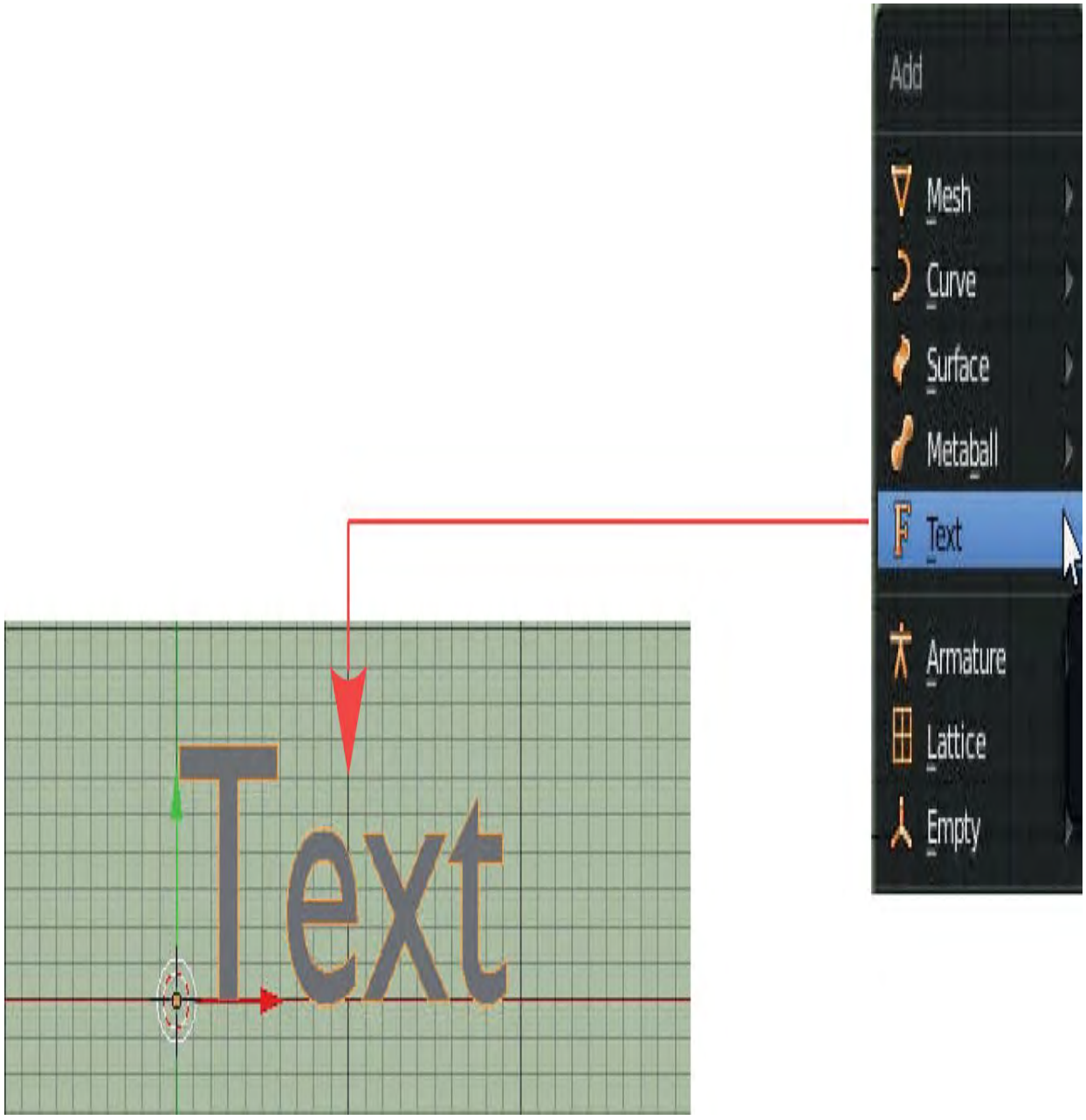


Figure 13.1

Tab into Edit mode—the word **Text** now has a Typing cursor at the end. (Figure 13.2).

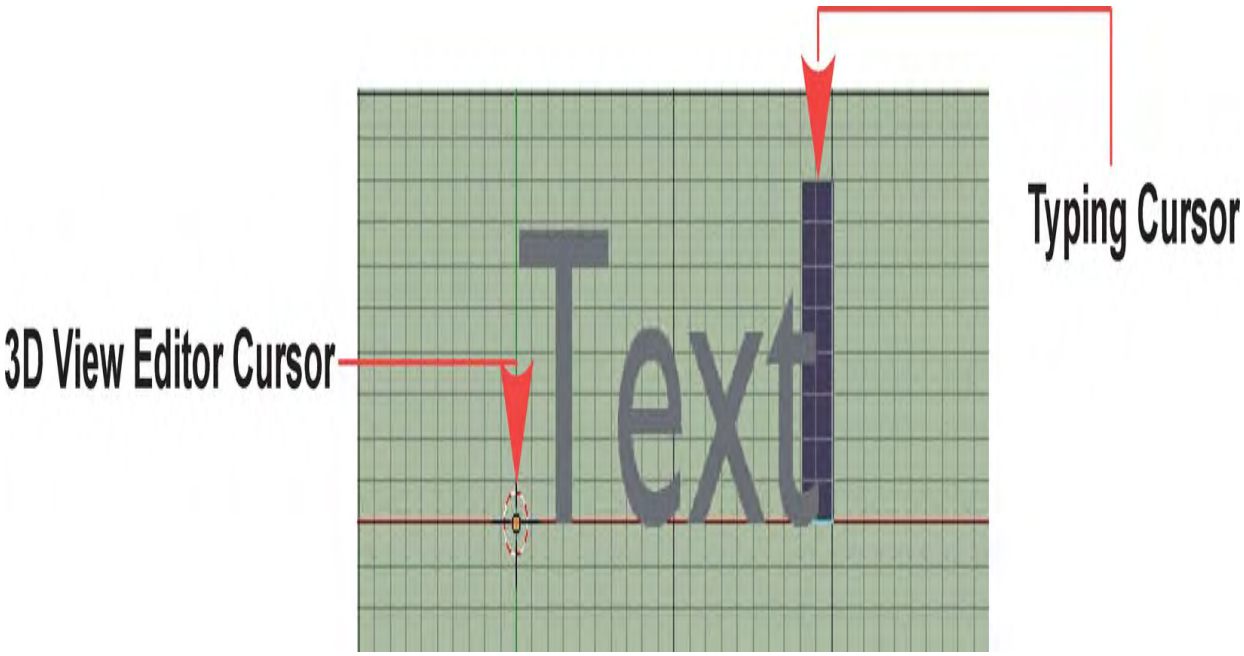
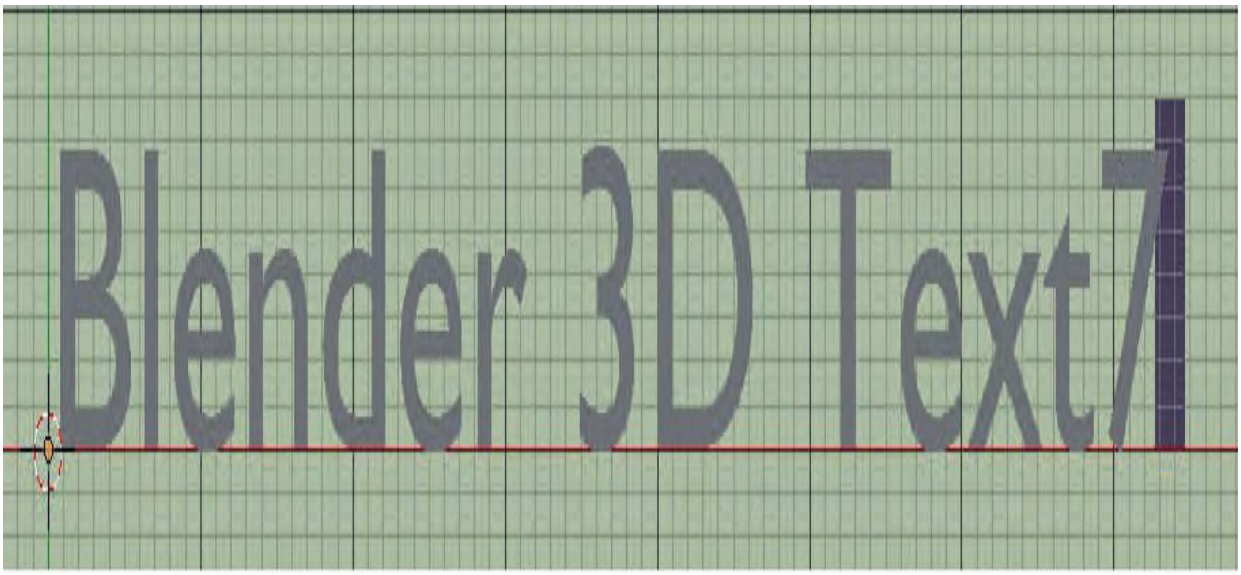


Figure 13.2

Backspace to delete letters and type in your own words just like in a text editor. Don’t worry about the font style or size at this stage. When you have typed in the words, tab back into Object Mode; to shape and color the text (Figure 13.3).



Edit Mode: The word Text has been modified by retyping.

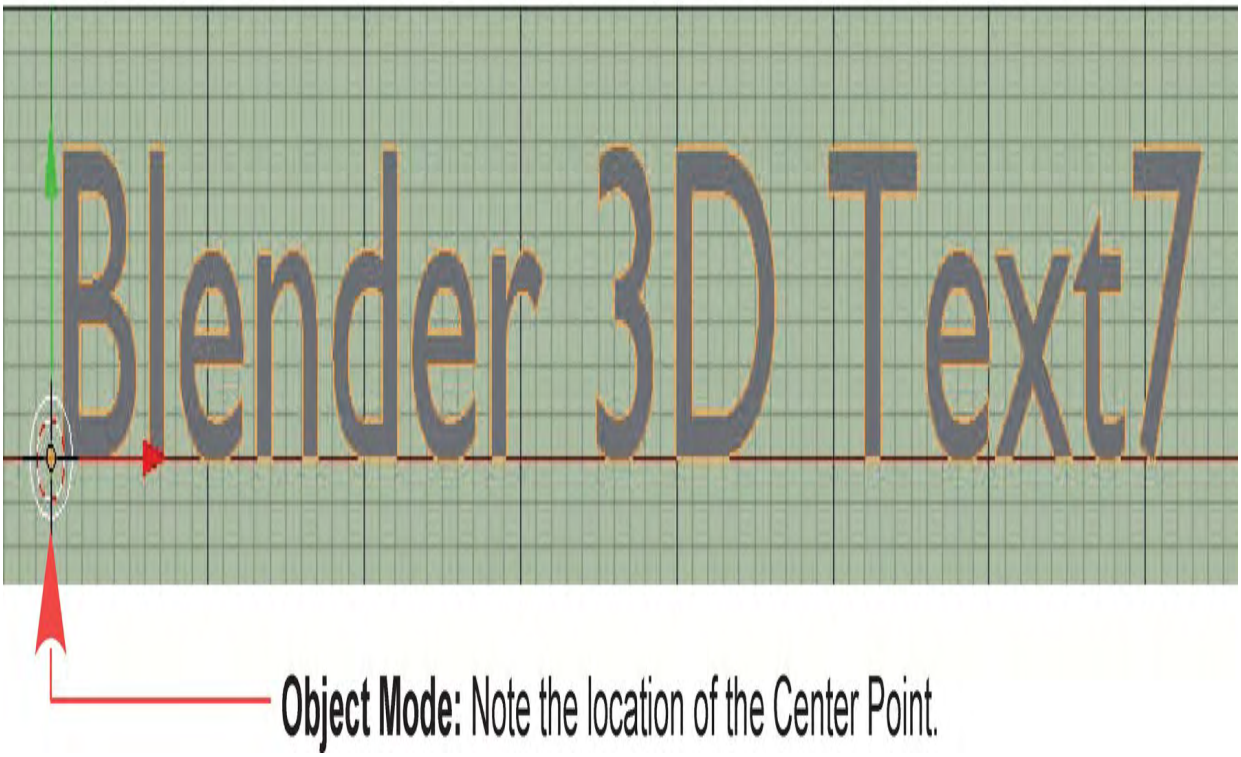


Figure 13.3

Concept – Text in Blender

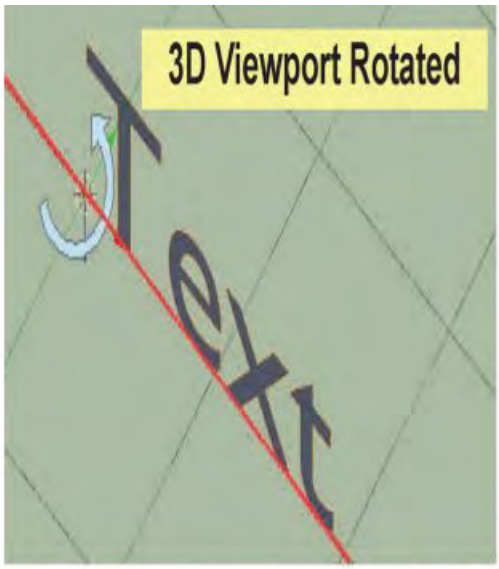
Typing Text in Blender is a little different to typing in a text editor program. When you Add Text in the 3D Viewport Editor, in Object Mode, the default word **Text** is an Object (a shape) similar to adding one of the Blender Primitives (Objects) such as a Cube, UV Sphere or Cone. When you entered Edit Mode, backspace and retyped a different word you modify the default Text Object. Besides modifying the letters in the text (the two dimensional shape of the characters) you can extrude thickness, bevel and round the shape. You can also add Material color, Texture and other effects. Modifying the text is done in the **Properties Editor.** Select the text in the 3D Viewport Editor, then go to the **Properties Editor, Object Data Properties.** Note: For Text the **Data button** is denoted by an **a.**

Properties Editor
Text
Text
Text
Shape
Resolution Preview U
12
Render U
0
Fast Editing
Fill Mode
Both
Fill Deformed
Texture Space
Geometry
Offset
0m
Extrude
0m
Taper Object
"a"
Bevel
Depth
0m
Resolution
4
Object
Fill Caps

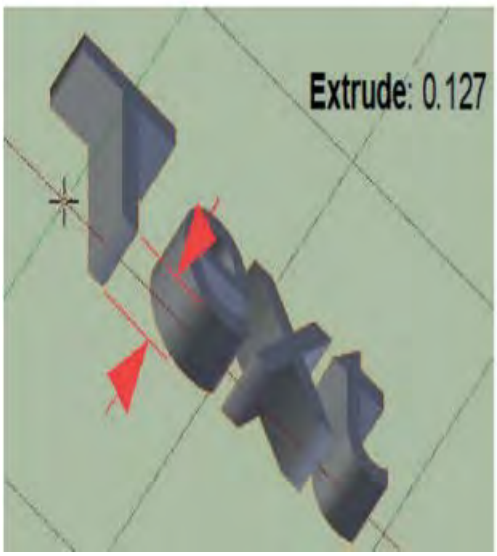
Figure 13.4

13.2 The Object Data Properties

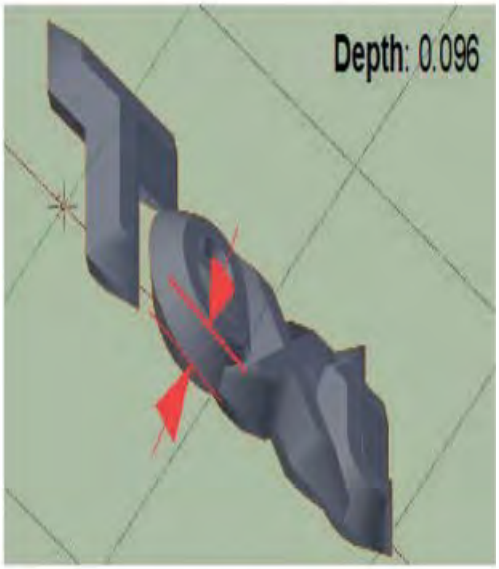
The Geometry Tab has settings for shaping the text into a three dimensional Object. To see the effect of the different settings rotate the text about the X Axis of the Scene then rotate the 3D Viewport as shown below.


3D Viewport Rotated
Text

Offset: 0.025
Text


Extrude: 0.127
Text

Bevel


Depth: 0.096
Text

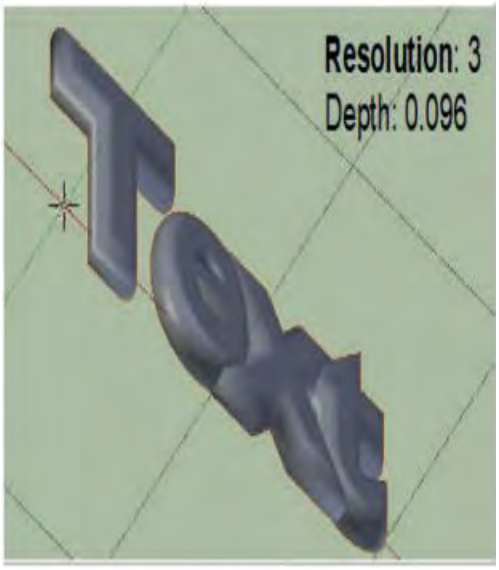

Resolution: 3
Depth: 0.096
Text

Figure 13.5

13.3 Fonts

The default **Font Style** is entered as **Bfont** as seen in the **Properties Editor, Object Data Properties, Font Tab. Bfont** is a **Vector Font for Text Objects** which is compiled into the Blender program and as such is not a standard font used in Windows or other operating systems. You can change the style to whatever font you have on your system. If you are using a Windows operating system, font styles can be found in **C:/Windows/Fonts.**

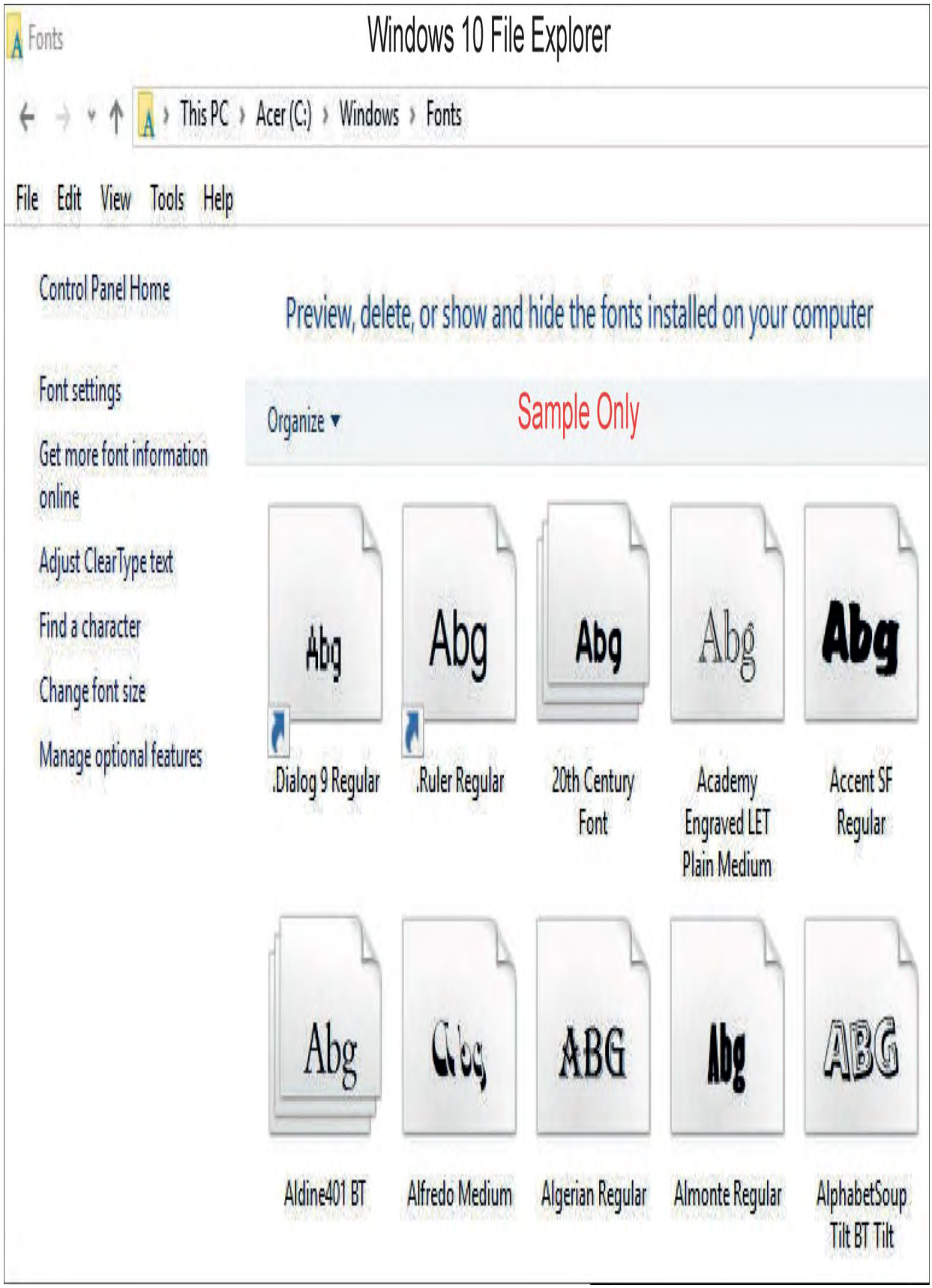

Fonts
Windows 10 File Explorer
This PC › Acer (C:) › Windows › Fonts
File Edit View Tools Help
Control Panel Home
Font settings
Get more font information online
Adjust ClearType text
Find a character
Change font size
Manage optional features
Preview, delete, or show and hide the fonts installed on your computer
Organize
Sample Only
.Dialog 9 Regular
.Ruler Regular
20th Century Font
Academy Engraved LET Plain Medium
Accent SF Regular
Aldine401 BT
Alfredo Medium
Algerian Regular
Almonte Regular
AlphabetSoup Tilt BT Tilt

Figure 13.6

For Example: To use Times New Roman Font Style.

Regular: times.ttf **Bold**: timesbd.ttf

Italic: timesbi.ttf **Bold & Italic**: timesi.ttf

Figure 13.7

To use Windows fonts in Blender they have to be entered in the font slots in the **Properties Editor, Object Data Properties, Font tab** (Figure 13.7).

> **Note:** In the Font tab there are four Font Style slots; Regular, Bold, Italic and Bold & Italic.
>
> A Font Style has to be entered for each slot.

For Example: To use Times New Roman Font St le.

All four **.ttf** Font Types have to be entered.

To enter a Font Style click on the **Folder icon** (Figure 13.8).

Figure 13.8

Clicking the Folder icon opens **Blenders File Viewer** where you navigate and locate the Font style (Figure 13.9). On a Windows system go to: C:\VVindows\Fonts\

Clicking the Folder icon opens **Blender's File Viewer** where you navigate and locate the Font style (Figure 13.9). On a Windows system go to: C:\Windows\Fonts\

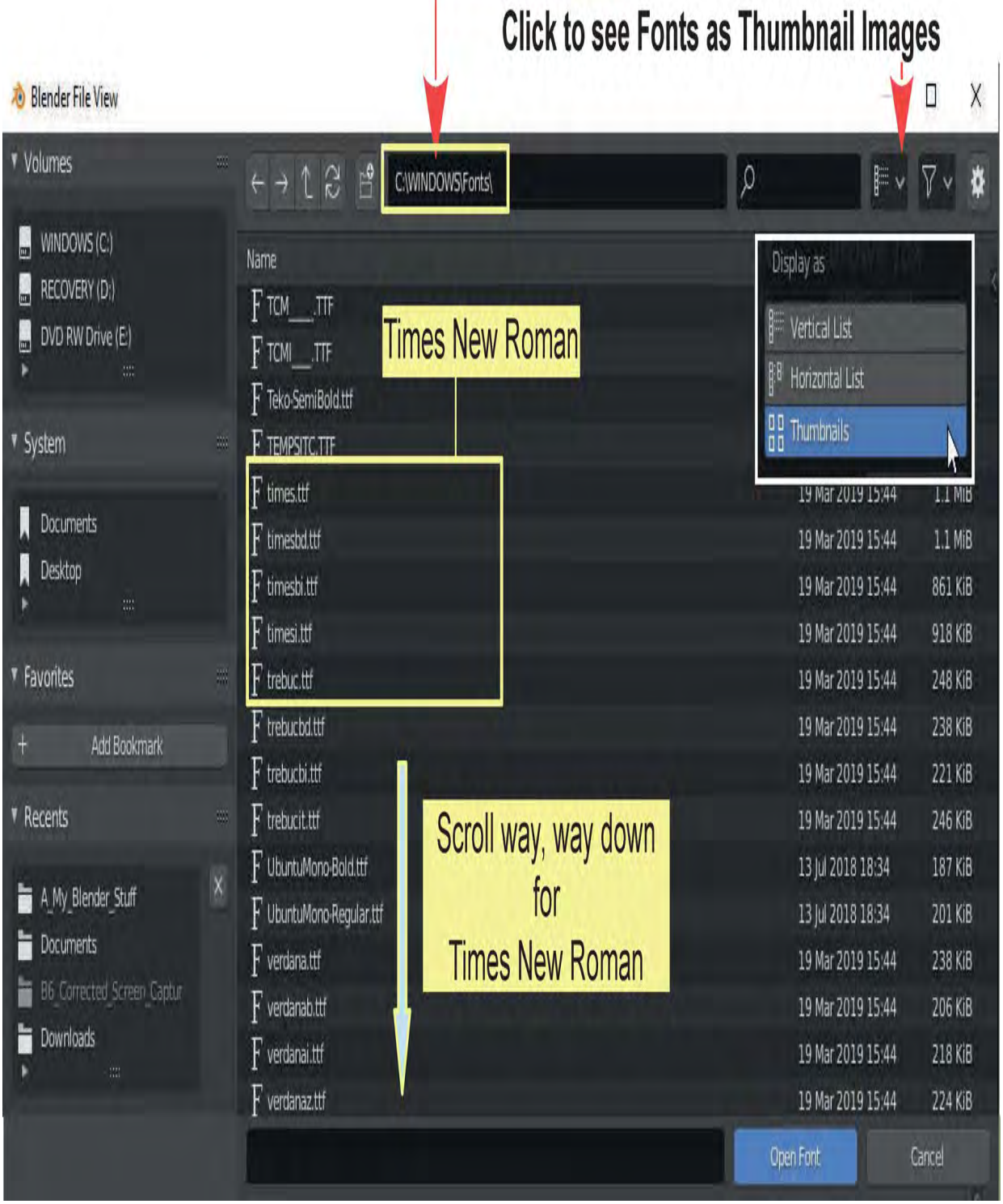


Figure 13.9

Note: Click in the Editor header to see the Fonts as Thumbnail Images.

Figure 13.10

Blender will accept any of the Windows fonts, but some may be distorted when they are extruded into 3D shapes.

Having selected a font click on the **Open Font** button in the lower RH corner of the File Viewer. Do this for each font slot.

Note: you can mix and match different font styles and options. For instance you can have one font for regular text and a different font for bold or italic.

Note: Entering Fonts in the Properties Editor, Fonts Object Data Properties, Font tab modifies the data for the particular Text Object you are working on in the 3D Viewport Editor (the object you have selected). It does not set the Font type for every time you add Text into the 3D Viewport Editor. When you add a new Text Object it will be the default Blender **Bfont.**

By default, typing will enter text in the 3D Viewport Editor in **Regular** text. To enter text in one of the alternative options (Bold, Italic, Underline, Small Caps) click one of the option **buttons** in the **Font Tab** (Figure 13.11). **Note: The options only display in Edit Mode.**

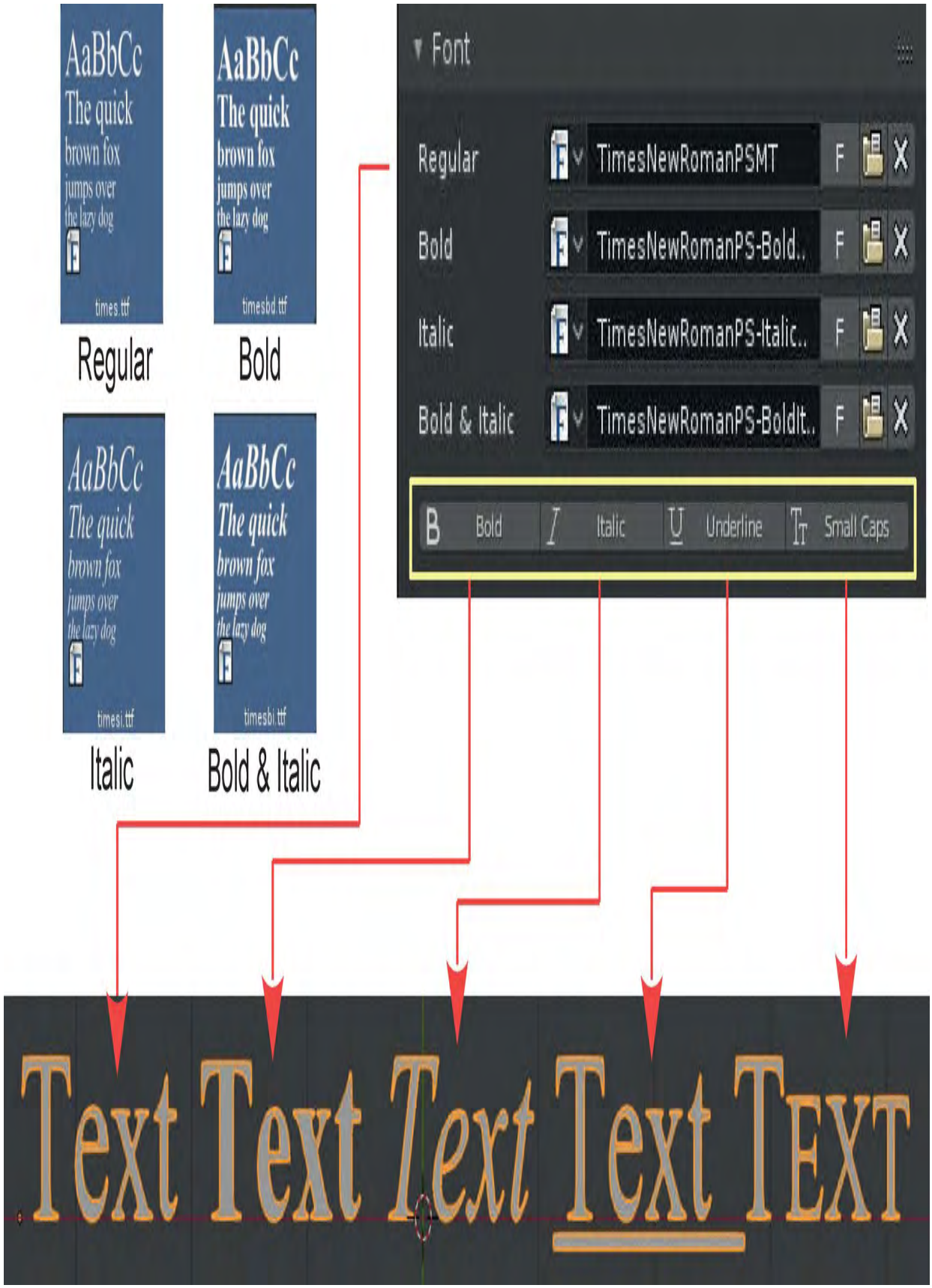

AaBbCc
The quick
brown fox
jumps over
the lazy dog
times.ttf
Regular
AaBbCc
The quick
brown fox
jumps over
the lazy dog
timesbd.ttf
Bold
AaBbCc
The quick
brown fox
jumps over
the lazy dog
timesi.ttf
Italic
AaBbCc
The quick
brown fox
jumps over
the lazy dog
timesbi.ttf
Bold & Italic
Font
Regular
TimesNewRomanPSMT
F
Bold
TimesNewRomanPS-Bold..
F
Italic
TimesNewRomanPS-Italic..
F
Bold & Italic
TimesNewRomanPS-BoldIt..
F
B Bold
I Italic
U Underline
Tr Small Caps
Text Text Text Text TEXT

Figure 13.11

To change text that has already been entered, locate the **Text Cursor** (in Edit mode) **using the arrow keys** on the keyboard, press and hold shift while using the arrow keys to highlight text, delete the text then check one of the **Character buttons** to change the text option. Retype the text with the new option. Pressing **Ctrl + L or R arrow keys** moves the text cursor to the end of a word.

In the **Font tab,** the underline position and thickness values only operate when **Underline** is ticked under the **Character** heading. Underlining occurs as you type your text in Edit mode.

Also in the **Font Tab,** adjusting the Size and Shear value sliders increases the size of the selected Text object and shears the Text (similar to Italic — Figure 13.12).

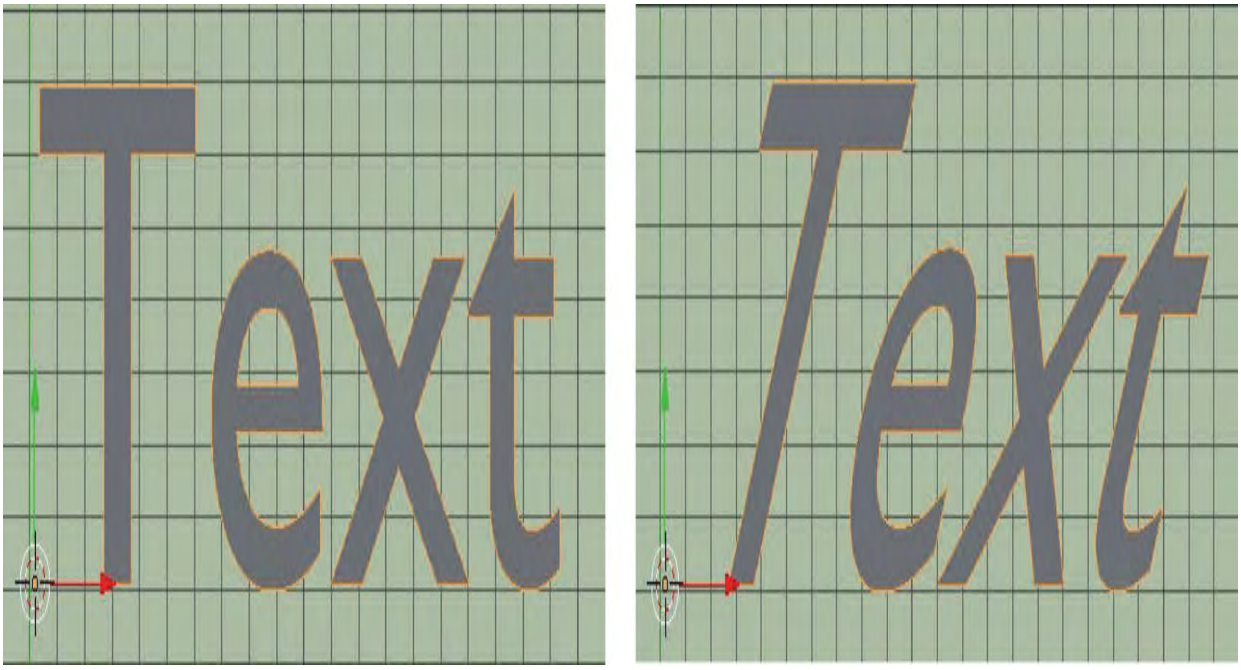


Figure 13.12

13.4 Text on a Curve

Text in Blender can be made to follow the shape of a Curved Path. Add text to your Scene as previously described then in Edit mode type something to extend the text (Figure 13.13).

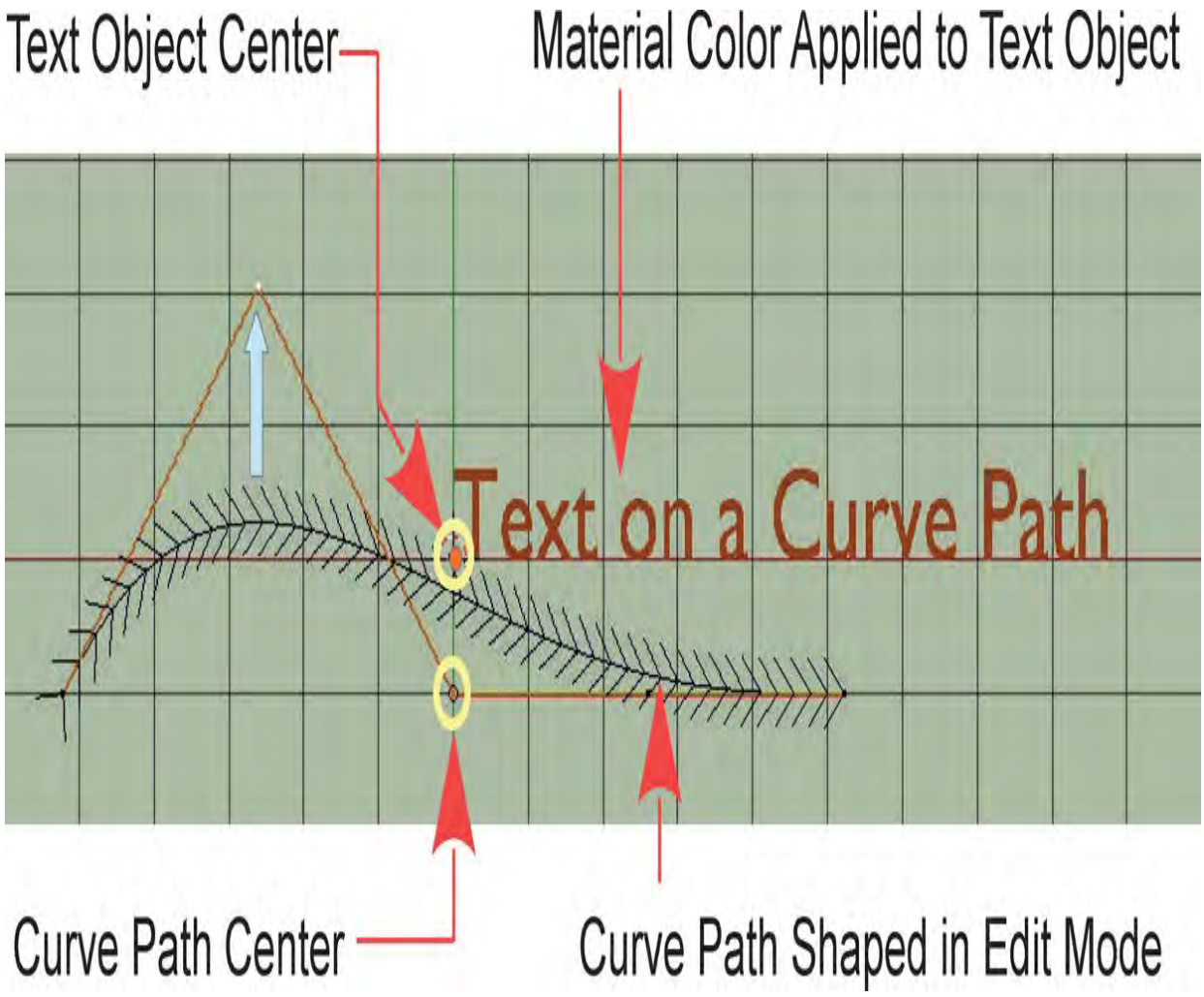


Figure 13.13

Add a **Curve - Path** to the Scene **(Shift + A key — Curve — Path).** Note: By default the path is named **NurbsPath.**

The Curve Path is added to the Scene in Object Mode and appears as a straight line. Scale the Path to make it longer and reposition it in Object Mode.

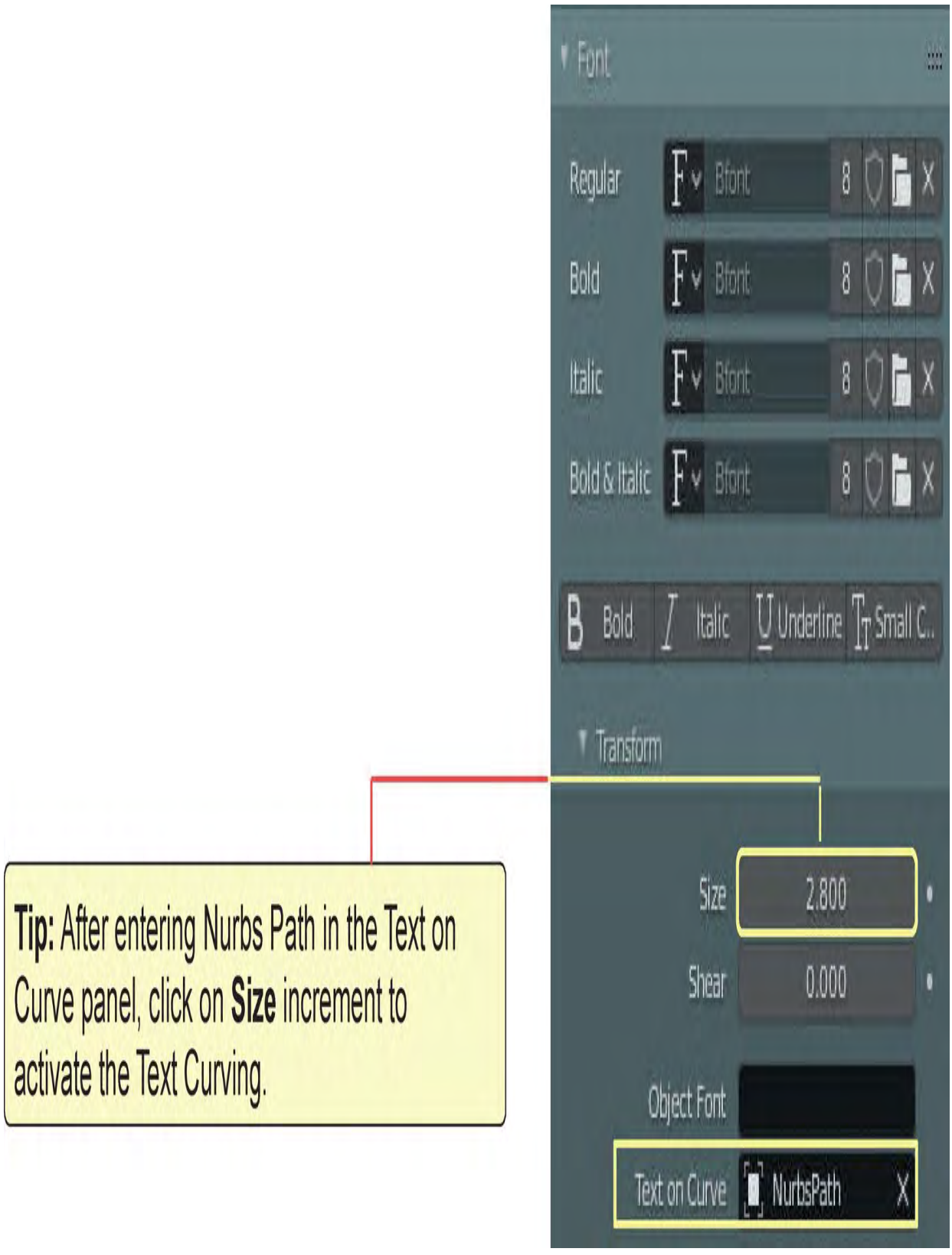

Font
Regular
Bfont
8
Bold
Bfont
8
Italic
Bfont
8
Bold & Italic
Bfont
8
Bold
Italic
Underline
Small C..
Transform
Size
2.800
Shear
0.000
Object Font
Text on Curve
NurbsPath
Tip: After entering Nurbs Path in the Text on Curve panel, click on Size increment to activate the Text Curving.

Figure 13.14

In Edit Mode shape the Curve (Figure 13.13). With the Curve shaped tab back to Object mode and deselect it.

Select the Text Object then in the **Properties Editor, Data Properties, Font Tab,** find the **Text on Curve panel** under **Transform.** Click on the little cube icon and in the drop down menu that displays select **NurbsPath** (Figure 13.14).

The text is shaped to follow the profile of the curve (Figure 13.15).

Tip: After entering Nurbs Path in the Text on Curve panel, click on **Size** increment to activate the Text Curving.

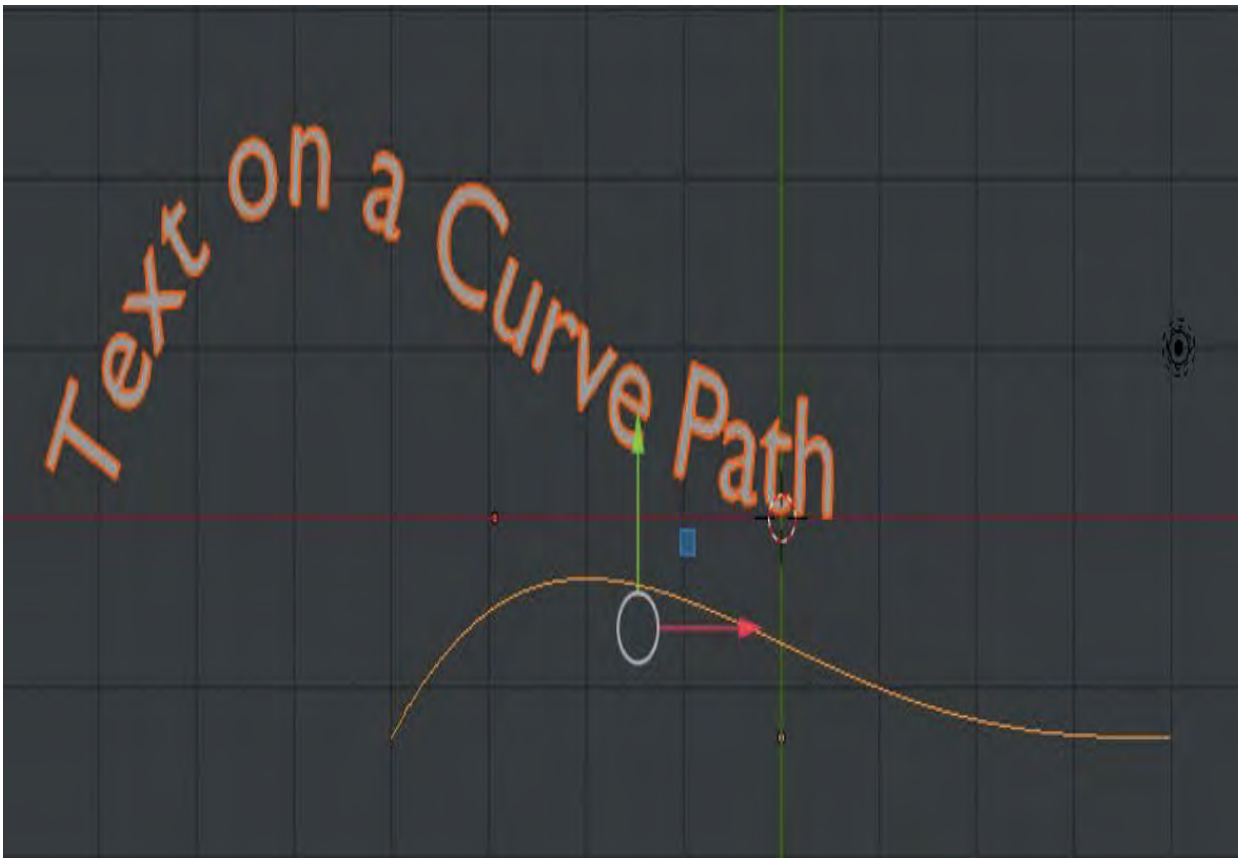
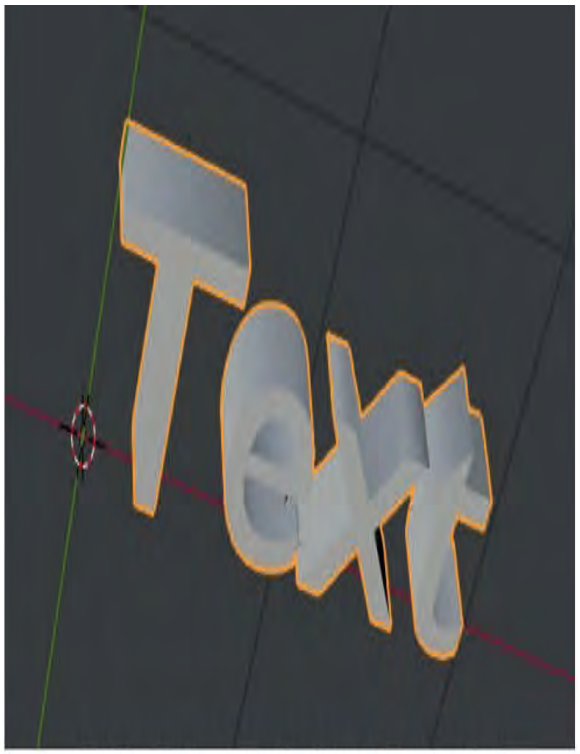


Figure 13.15

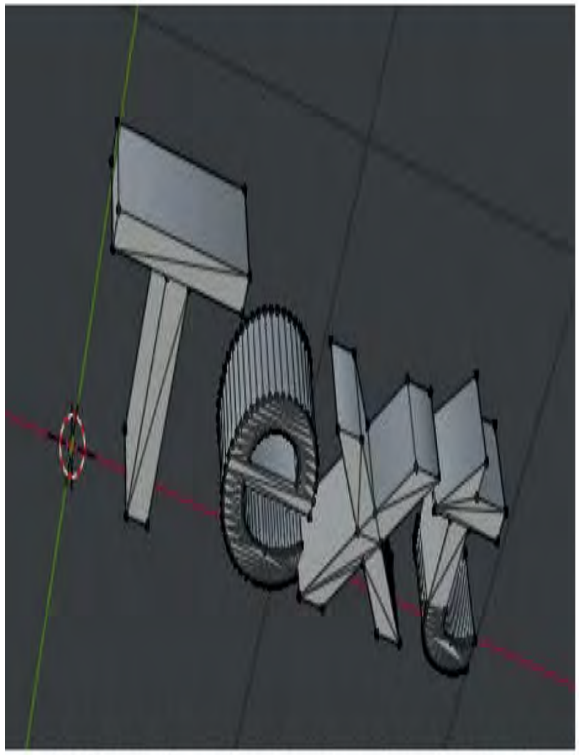
13.5 Converting Text to a Mesh Object

There is only limited functionality in the text **Object Data Properties, Geometry tab** for modifying the text shape (see 13.1 Figure 13.4).

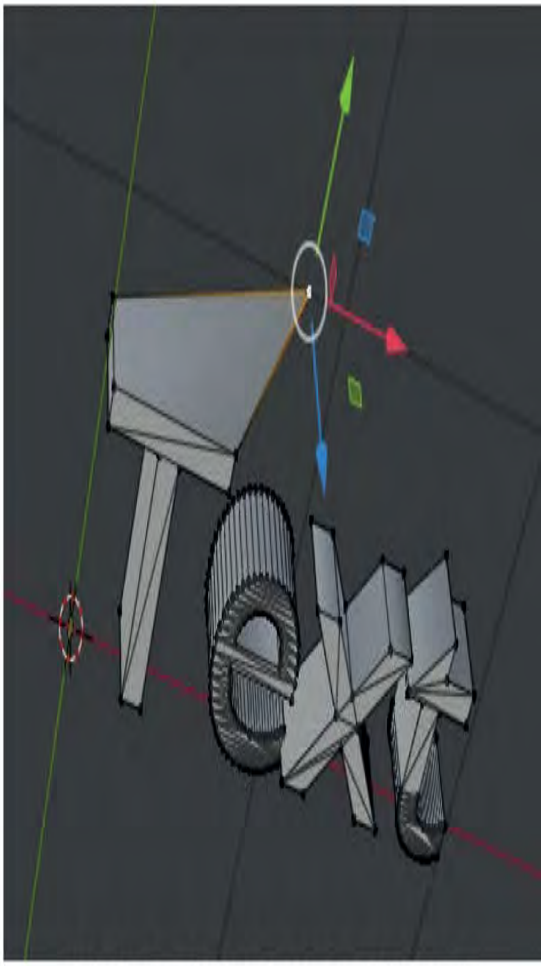
When you add Text to a Scene it remains a 2D Plane Object unless you have extruded the text in the Geometry tab. Entering Edit mode only allows you to retype a text change. To perform editing, which actually changes the detailed shape of the Text, you have to convert to a **Mesh Object.** To do this, select the text in **Object Mode** then in the **3D Viewport Editor Header** click **Object - Convert - Mesh.** Tab to Edit mode and you will see that the text is now a Mesh Object with vertices that can be moved, rotated, and scaled (Figure 13.16).



Object Mode with Text Extruded



Edit Mode after Conversion to Mesh



Vertices Selected and Moved

Figure 13.16

13.6 Converting Text to a Curve

If you would like to perform some fancy editing of a single letter, you can convert the letter into a Curve. The outline of the letter becomes a Curve with handles, which allows you to manipulate the shape into anything you wish.

Add Text, then in Edit Mode backspace until you are left with the single letter T . Scale, rotate, and move it where you like then tab to **Object Mode.** In the **3D View Editor Header** click **Object- Convert — Curve** (Figure 13.17). In **Edit Mode** you will see the your letter with what appear to be Vertices. By placing the Mouse Cursor (White Cross) in proximity to a Vertex and clicking LMB you display Control Handles at that Vertex. Click LMB on one of the Control Points to select, G Key (Grab) and move to shape the Text profile (Figure 13.18). Note: In Edit Mode you may press A Key to select and show all the Control Handles then RMB Click and Subdivide creating additional handles.

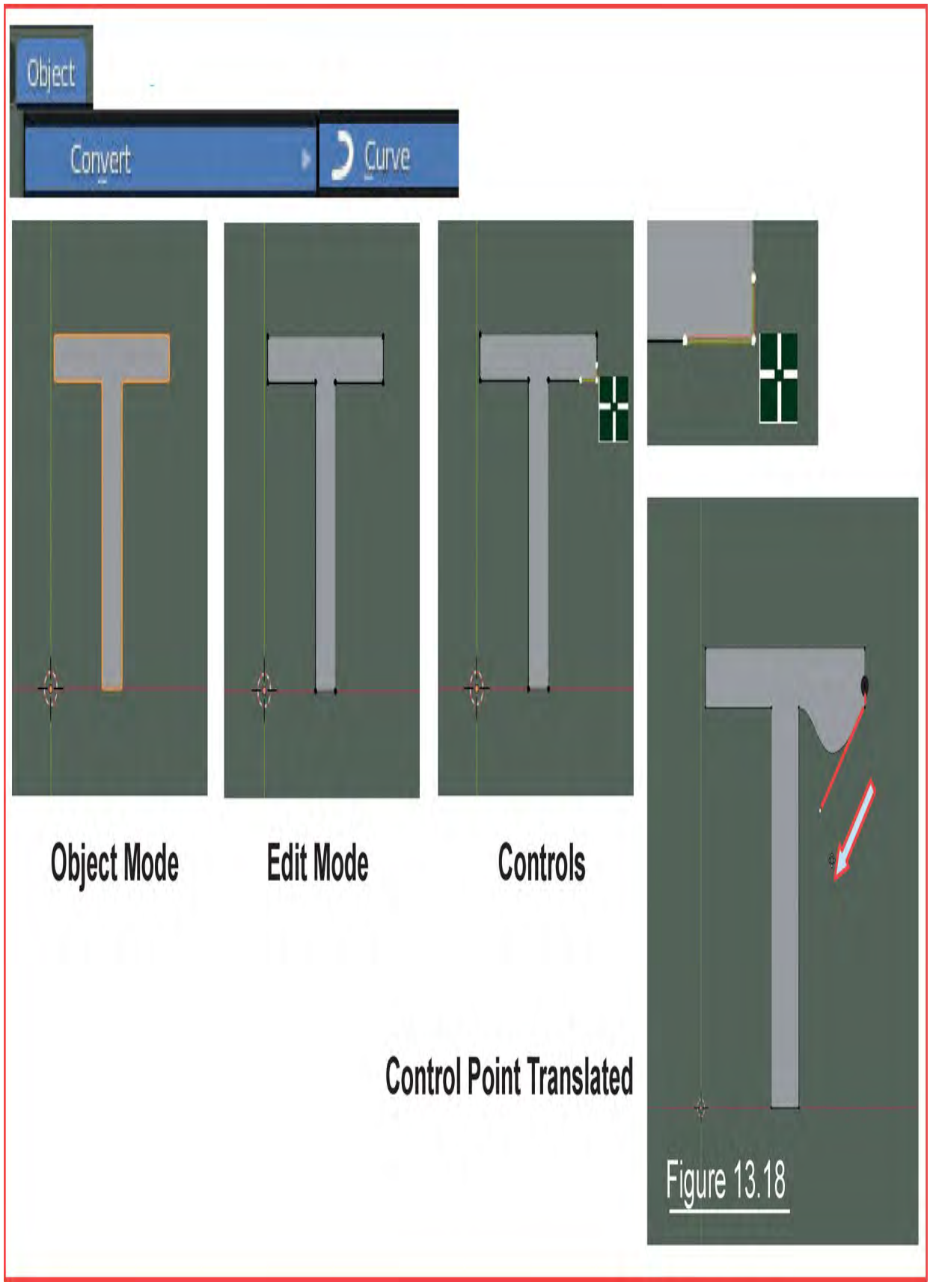

Object
Convert
Curve
Object Mode
Edit Mode
Controls
Control Point Translated

Figure 13.18

Figure 13.17

13.7 Entering External Font

Text created in a Text Editor such as Word Pad or any editor that saves a file in .txt format may be entered into Blender.

For example a **Text File** named **Test_Text.txt** created in **Word Pad,** using **Font Style: Courier New size 36** saved in **MyDocuments** folder. The file contains the single word **Texting.**

To enter this in Blender, click **Add - Text** in the **3D Viewport Editor Header** (alternatively press **Shift + A key** and select **Text** from the menu.

Remember have the **3D Viewport in Top Orthographic** View.

Tab to **Edit mode** and backspace deleting the default word **Text.**

In the **3D View Editor Header** click on **Edit** and select **Past File** from the menu. The File Browser window opens where you navigate and find the saved .txd file (Figure 13.19 over).

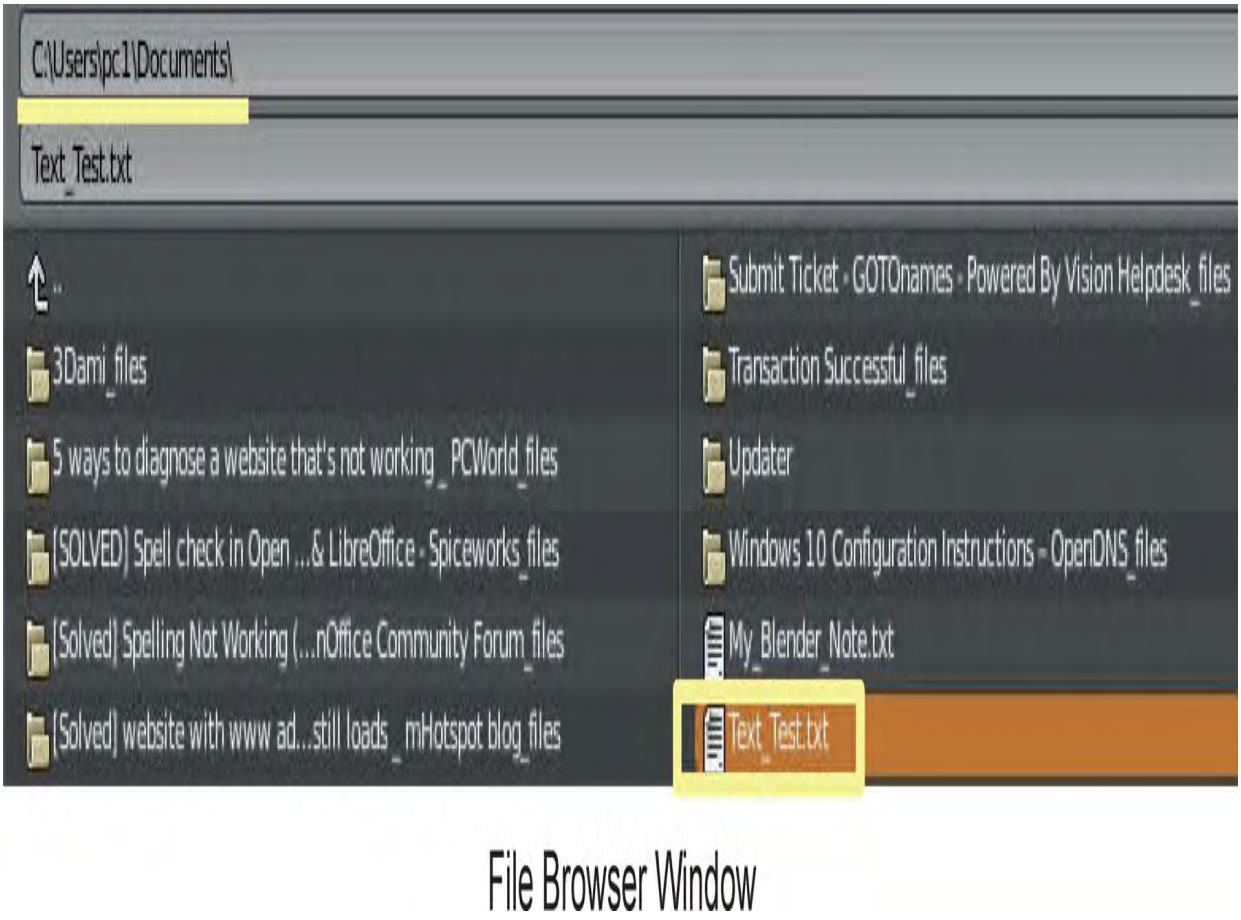


Figure 13.19

Click on the file name to highlight then click on **Paste File** in the upper **RH** corner of the window.

The text **Texting** is entered into the 3D View Editor in Edit mode (Figure 13.20).

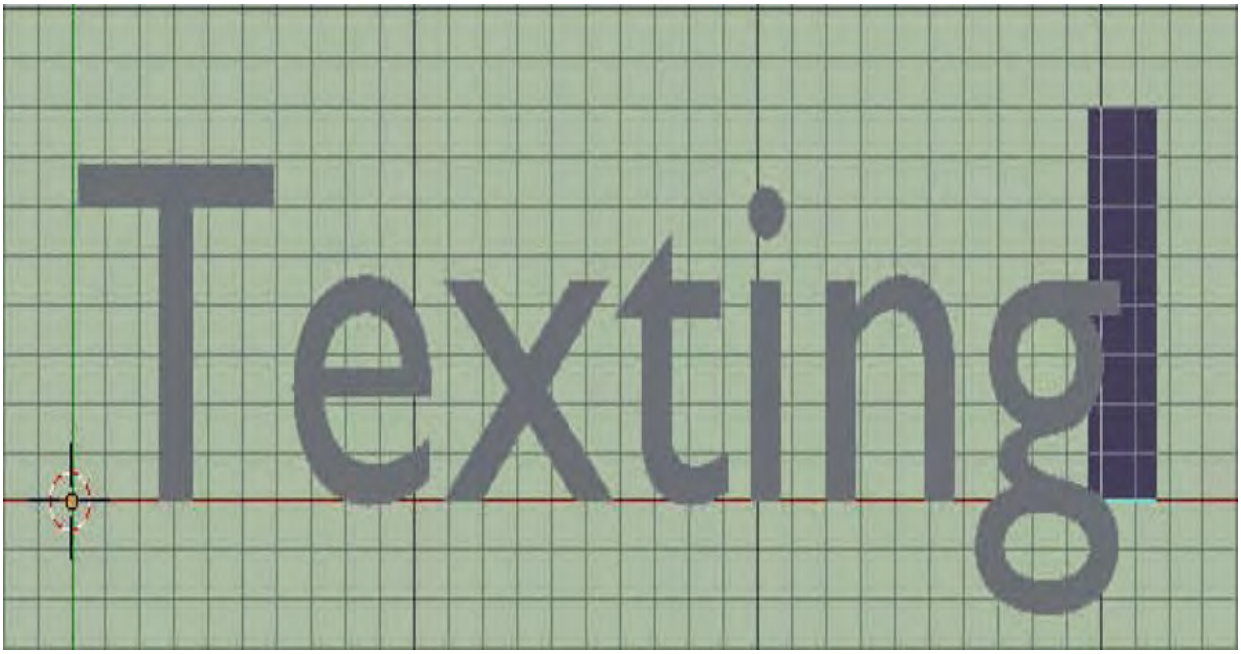
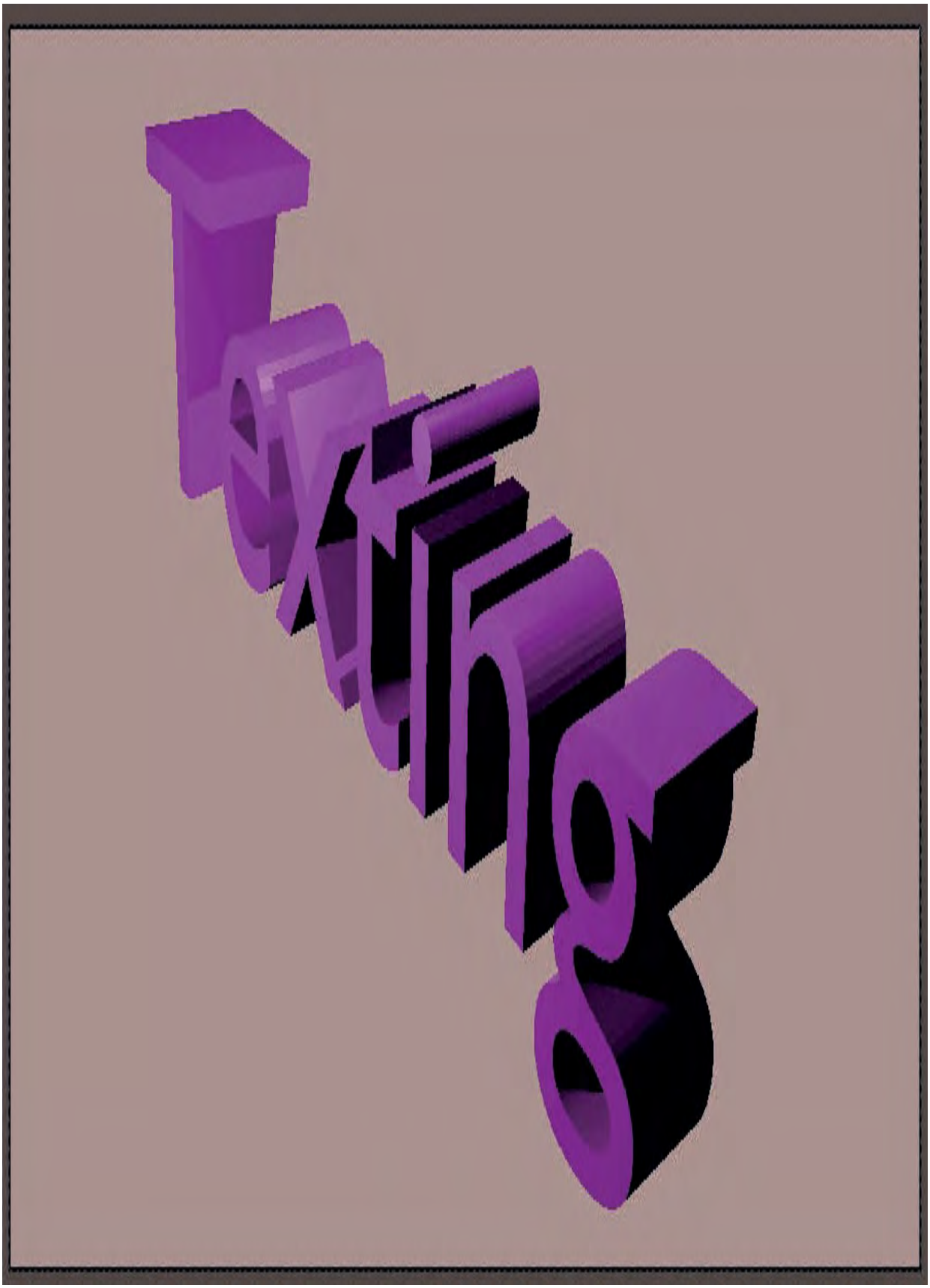


Figure 13.20

You may now convert the text to a **Mesh object** (see 13.5) or to a **Curve** (see 13.4).


Lighting

14 Viewport Shading

14.1 Viewport Shading Options
14.2 Wireframe Viewport Shading
14.3 Solid Viewport Shading
14.4 Color Display Options
14.5 Background Displays
14.6 More Solid Viewport Shading
14.7 Rendered Viewport Shading
14.8 Material Preview Shading
14.9 World Settings
14.10 Simplified Viewport Shading

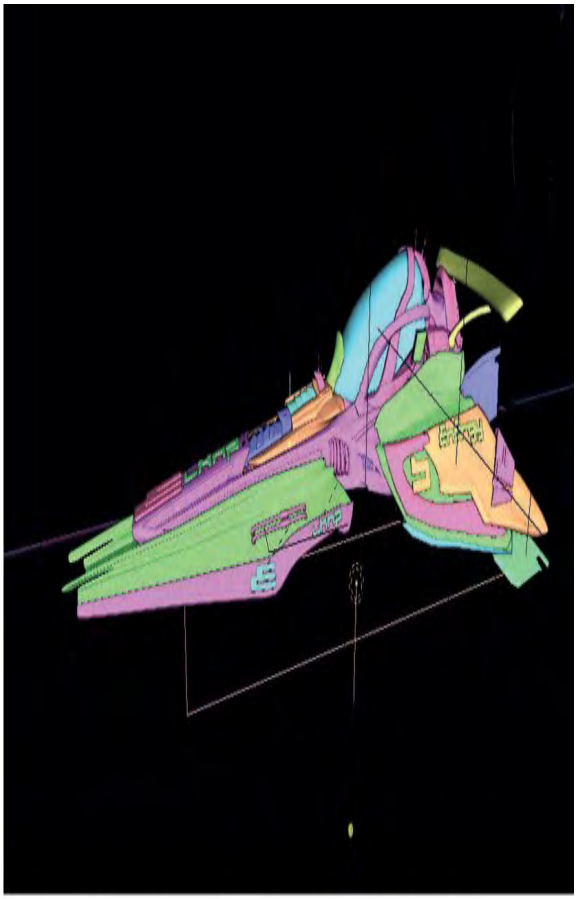
Viewport Shading

Definitions: Viewport: The View in an Editor. **Shading:** How the View in an Editor is displayed.

Blender provides a variety of shading options which allows a Scene to be previewed during construction. Previewing, in turn, allows editing to be performed as the Scene is developed.

As Scenes are created, with Objects being added and Materials (colors) applied and lighting effects and textures introduced, it can be difficult to isolate a particular Object or even a component of an Object. By shading the **Viewport,** making the Objects display in a simplified way, allows a selection to be Edited. You may also preview how a Scene will display before adding too much detail.

Rendered Viewport Shading



Simplified Viewport Shading

14.1 Viewport Shading Options

Options for **Viewport Shading** (displaying the 3D Viewport Editor) are found in the upper RH corner of the 3D Viewport Editor Header (Figure 14.1). Click LMB on one of the buttons to activate an option.

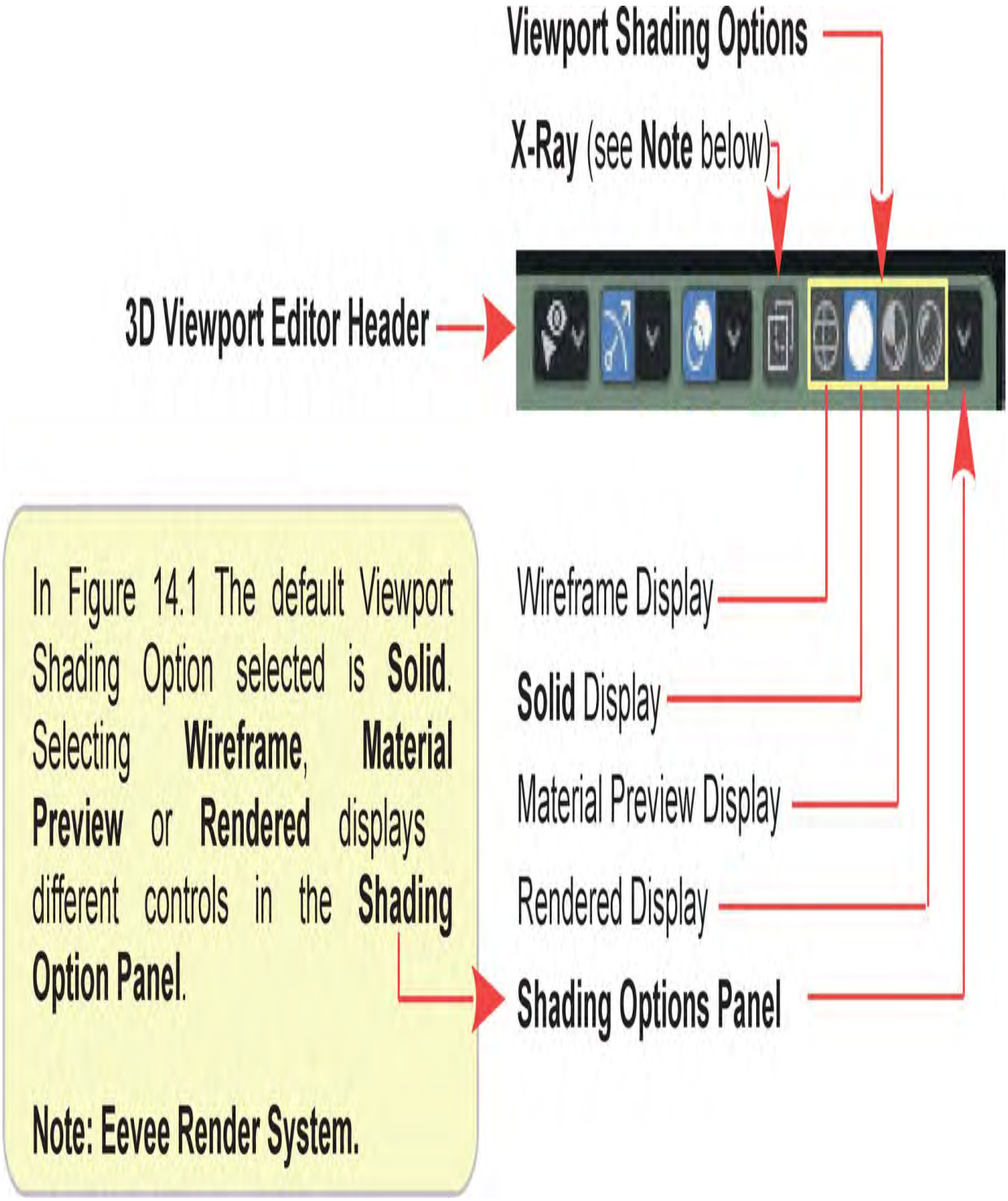


Figure 14.1

Note: The controls in the **Shading Option Panel** are applicable to the Viewport Shading Option that is selected in the 3D Viewport Editor

Header.

14.2 Wireframe Viewport Shading

Wireframe Shading allow you to see Objects in the 3D Viewport Editor as a **Mesh** which is useful when one Object overlaps or hides another. In Figure 14.2 a Monkey Object is positioned inside a UV Sphere. In Solid Viewport Shading Mode you see the outline of the Monkey **when the Monkey is selected.** In **Wireframe Viewport Shading Mode** you see both Objects as a Mesh with the selected Object displayed orange (Figure 14.2)

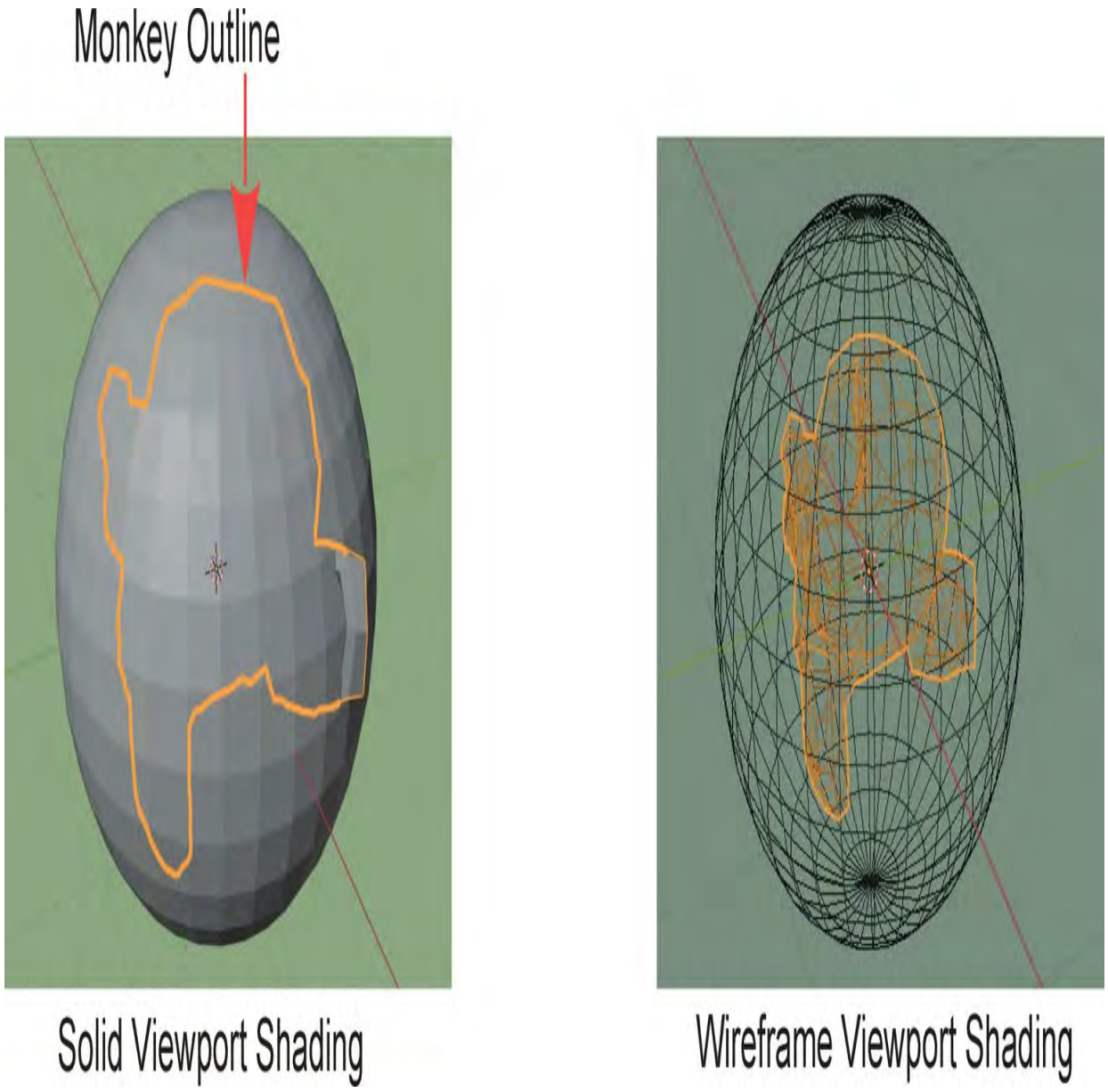


Figure 14.2

Note: When Wireframe Display Mode is activated, **X-Ray** is also activated allowing you to see details of the Mesh behind the front surface. You may turn X-Ray off if you wish.

Wireframe Colors: The Wireframe Mesh may be set to display in different colors for each Object. This is helpful when arranging Objects in groups. The

Mesh display colors are independent of the Material color applied to an Object.

A detailed study of Object Materials (colors) is explained in Chapter 16. In Chapter 1 – 1.12 coloring Objects was demonstrated by disabling the Blender Node System. This allowed Objects to be colored and the color viewed in Solid Viewport Shading Mode in the 3D Viewport Editor.

At this point, the UV Sphere and the Monkey Objects have been entered in the Scene but they **have not** had a Material applied. They display gray in Solid Viewport Shading Mode in the 3D Viewport Editor since they are using Blenders default Material **(a Material HAS NOT been applied).** If you select each Object in turn and look at the **Properties Editor, Material Properties** you see only the **New** button.

To have the Wireframe Mesh display in a color, obviously, be in **Wireframe Display Mode.** As an example, select the Monkey by clicking on Suzanne in the Outliner Editor. By default Suzanne's Mesh displays orange (Figure 14.3). In the **Properties Editor, Object Properties, Viewport Display Tab,** click on the Color bar and select a color in the Color Picker Circle.

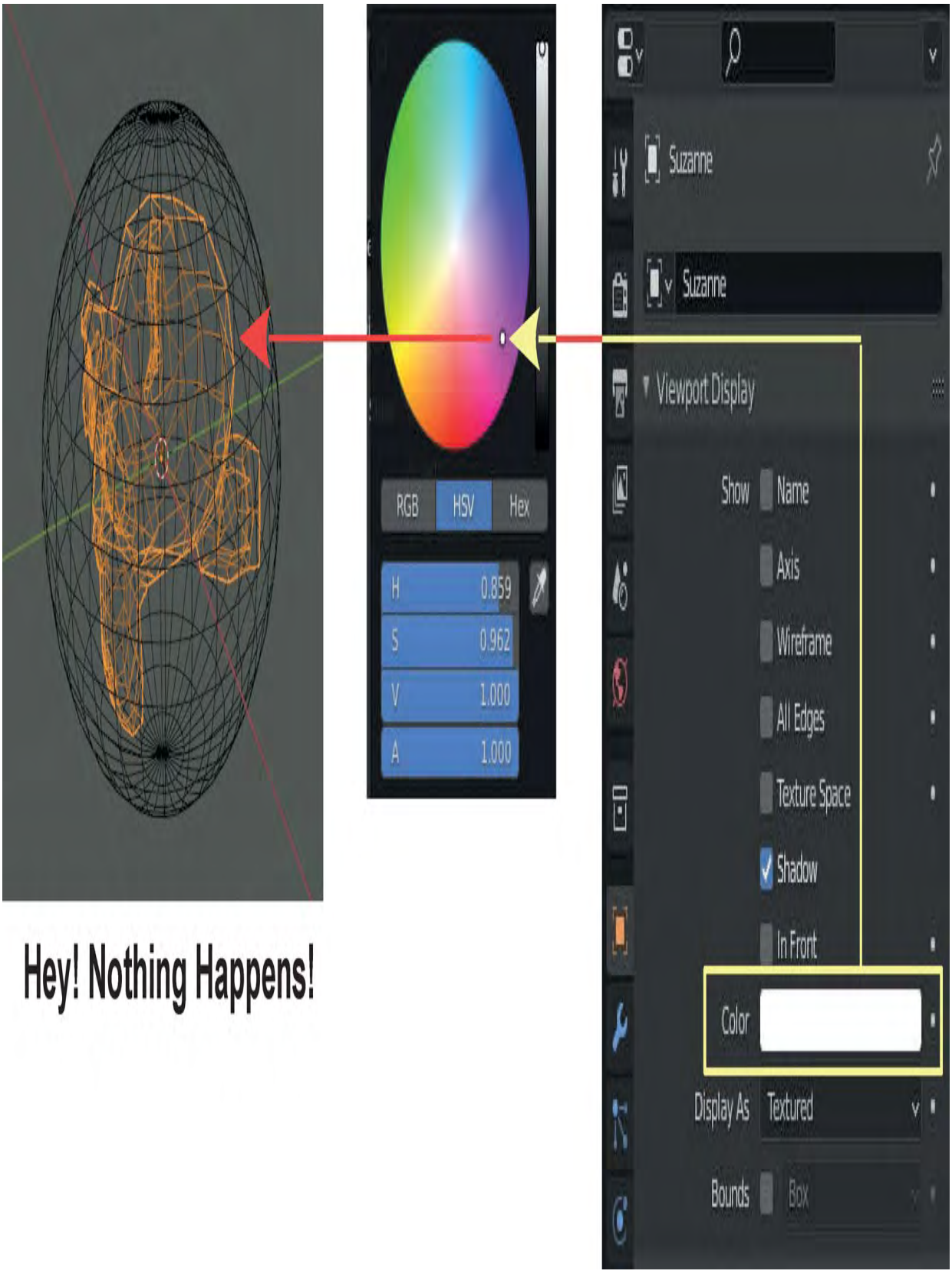

Hey! Nothing Happens!
RGB
HSV
Hex
H 0.859
S 0.962
V 1.000
A 1.000
Suzanne
Suzanne
Viewport Display
Show
Name
Axis
Wireframe
All Edges
Texture Space
Shadow
In Front
Color
Display As
Textured
Bounds
Box

Figure 14.3

Go to the **Shading Options Panel** for Wireframe display and change the Color setting to **Object**

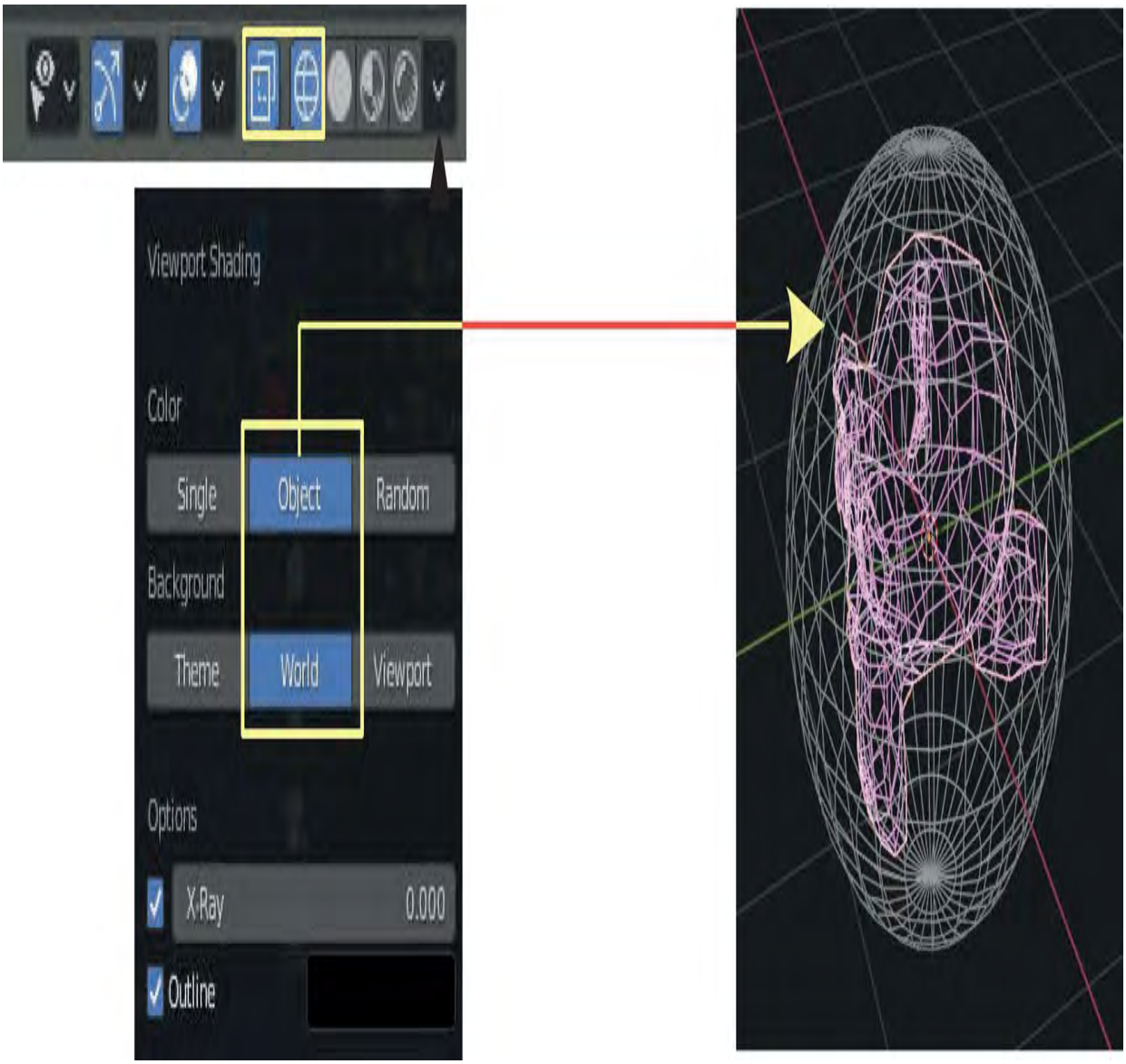


Figure 14.4
Figure 14.5

If you also change **Background** from **Theme** to **World** the 3D Viewport background provides a better contrast.

Changing the Background from Theme to World means you are changing the 3D Viewport Editor background color from the default Blender Theme set in the Preferences Editor (see Chapter 2 – 2.18) to the World, Surface Color, set in the Properties Editor, World Properties, Surface Tab.

A practical application of using VVireframe Mode is to position Objects inside a Cube creating an interior Scene.

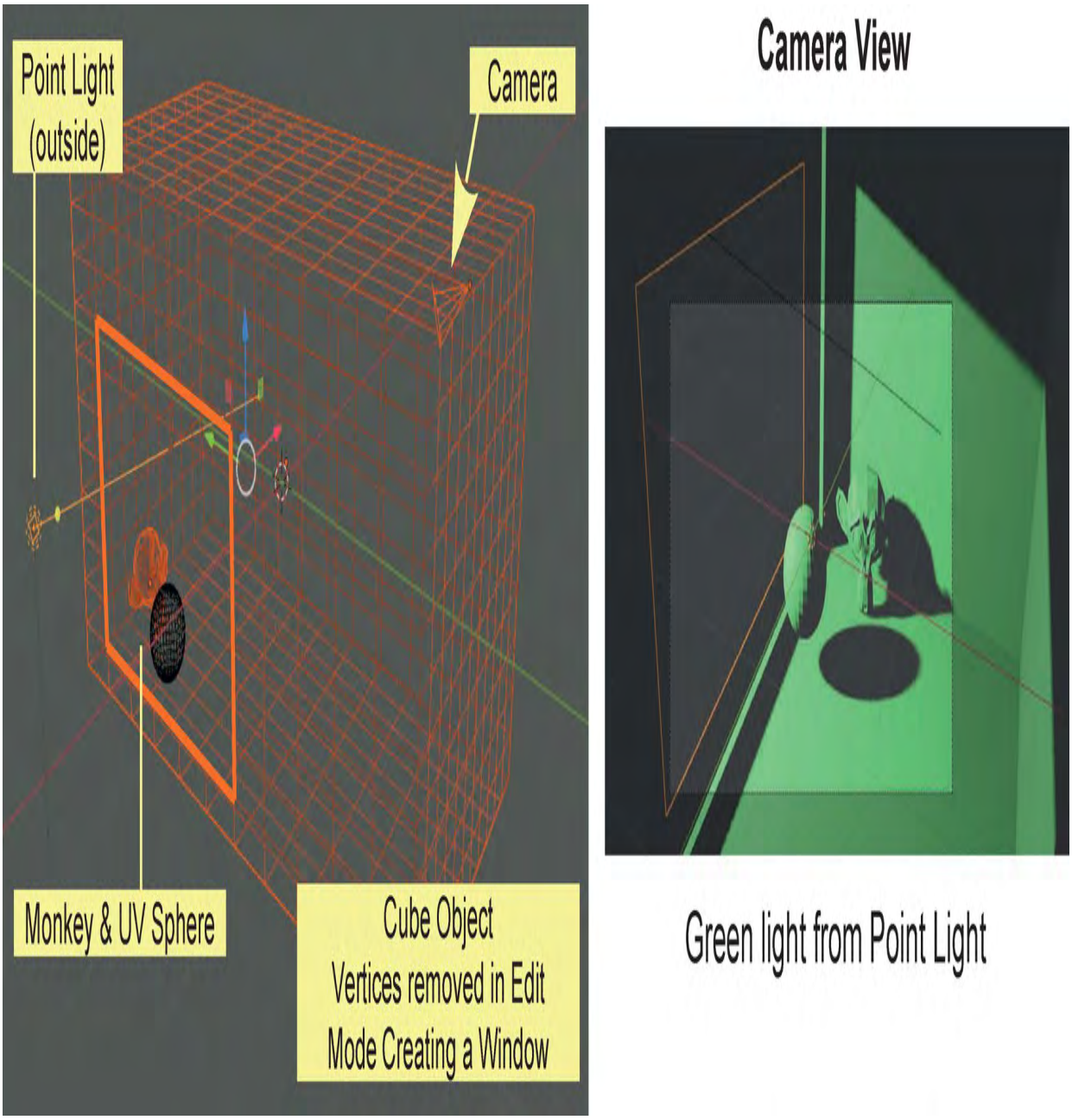


Figure 14.6

14.3 Solid Viewport Shading

Solid Viewport Shading is the default Shading Option for the 3D Viewport Editor when Blender Opens and is the basic Viewport display for modelling.

Sub Options Panel

Figure 14.7

The **Sub Options** for **Solid Viewport Shading** are categorised into four sections; **Lighting, Color, Background** and **Options** (sub, sub options).

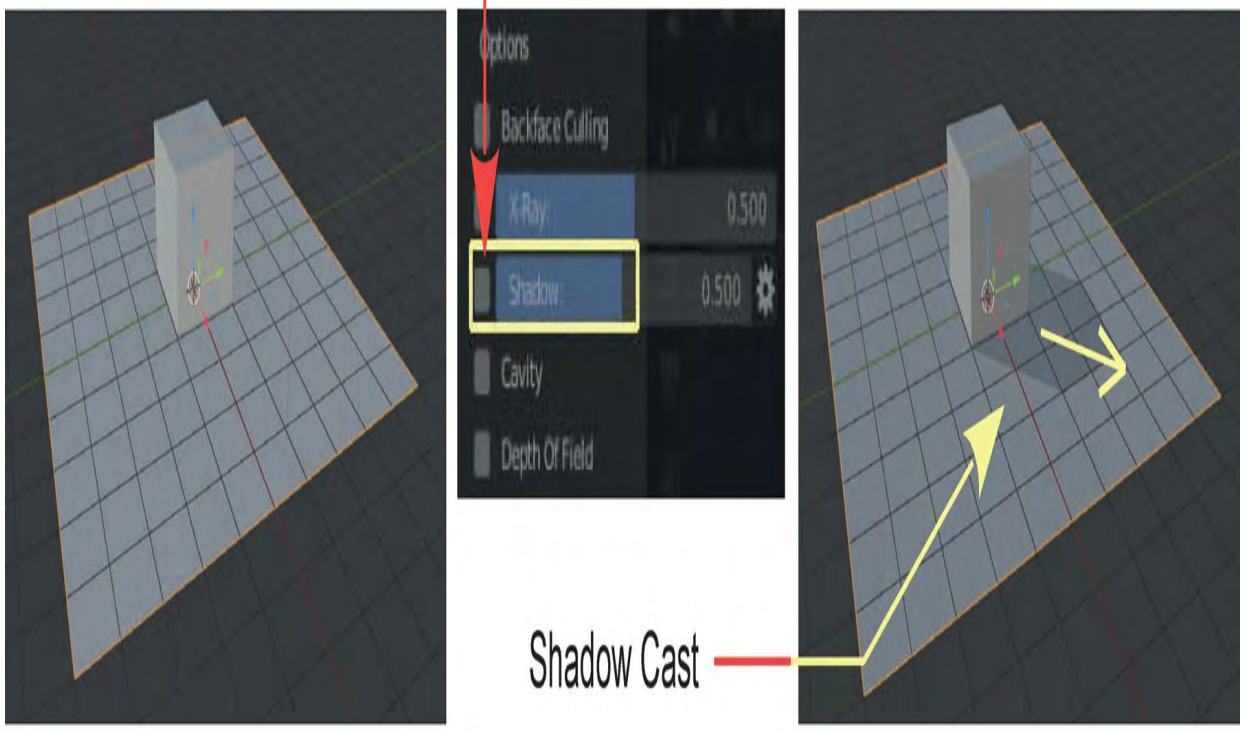
Lighting: A Scene is illuminated by placing Lights at strategic positions or having Emitter Objects (Objects that emit light) located thorough the Scene (Chapter 15). In the course of construction complicated Scene Lighting can be a hindrance, therefore, a simplified Viewport display is preferable.

Solid Viewport Shading provides three lighting methods; **Studio, MatCap** and **Flat.** These lighting methods are all independent of any Lamps placed in the 3D Viewport Editor. While in Solid Viewport Shading Mode Lights in the 3D Viewport Editor have no effect. **Studio Lighting** is the default method.

Studio Lighting is an arbitrary lighting arrangement which is independent of the Lights in the Scene. When the Viewport is rotated mesh faces are shaded for visualisation but the shading is not influenced by light sources such as the Lights in the Scene. Rendering the Scene produces a different shading based on the lighting generated by the Lights.

To examine and understand the relationship of Object illumination, shadows can be used. To see the effect of shadows there has to be something on which the shadows will be cast. In the default Scene add a Plane Object. The Plane is positioned on the Mid Plane Grid and scaled up. Position the Cube Object just above the Plane (Figure 14.12). With Solid Viewport Shading active, in the **Sub Options Panel, Options** (at the bottom of the panel) **check Shadow.**

check Shadow.



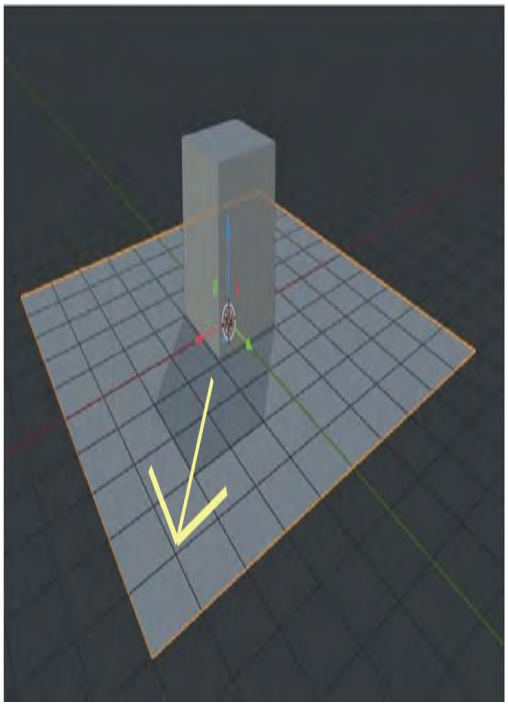


Figure 14.8

With the Scene rotated the shadow is cast relative to the Cube demonstrating that the light source is fixed relative to the Scene and is not relative to the Point Light.

In the Figure 14.8 the Cube and the Plane do NOT have Material (color) applied. To continue with the demonstration add color (Reference Chapter 4 - 4.18). Figure 14.9 shows Studio Lighting with color added.

By clicking on the sphere below Studio, MatCap Lighting Tones display. Clicking either of these alters the lighting tone in the 3D Viewport Editor.

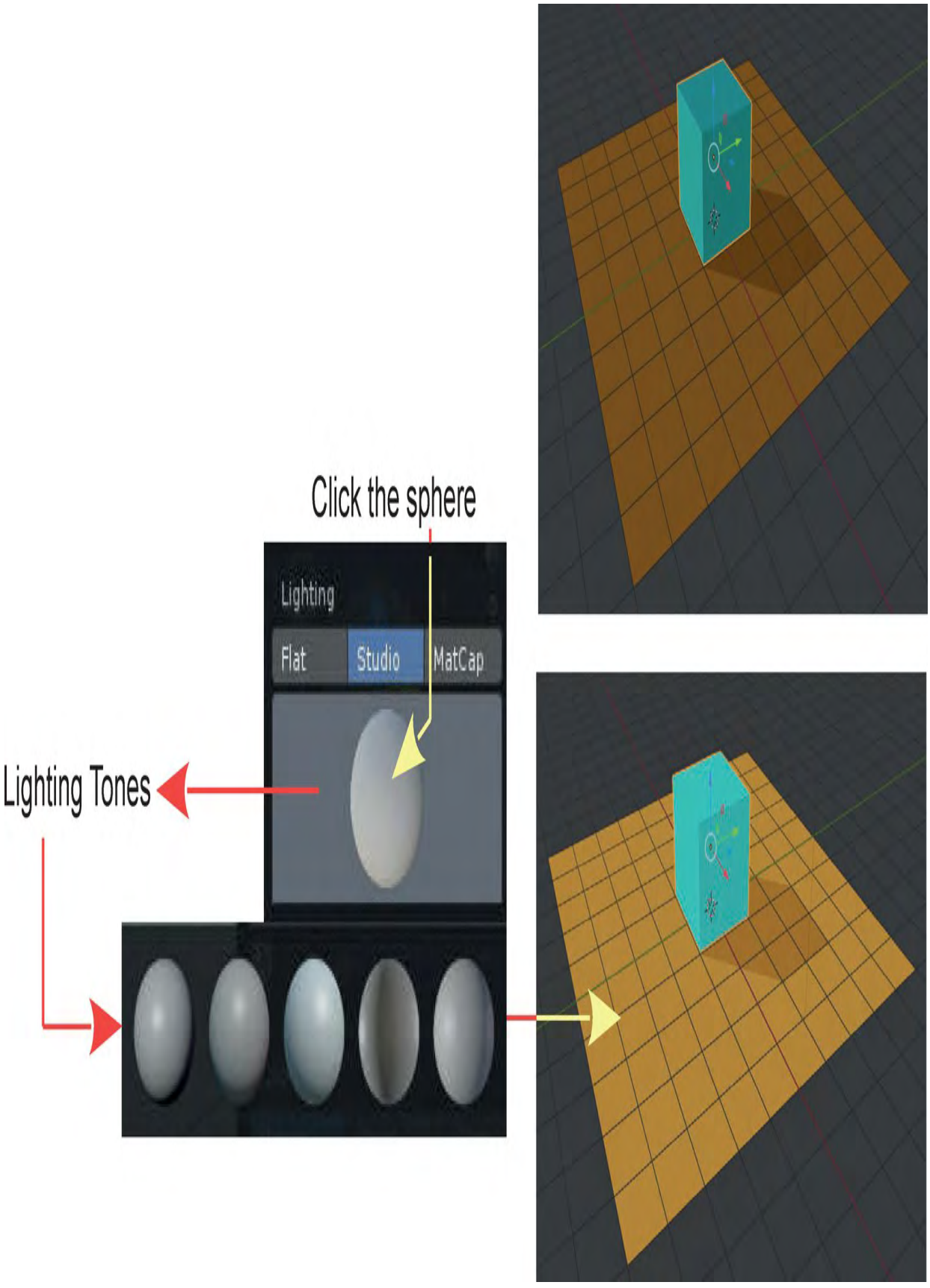

Click the sphere
Lighting
Flat
Studio
MatCap
Lighting Tones

Figure 14.9

Click LMB on the sphere to preview **Lighting Tones.**

Matcap Lighting stands for **material capture** – it is a complete material including lighting and reflections. A Matcap is added for quick feedback, to see how an Object's shape is changing under different lighting conditions. This is a preview only, not a permanent lighting set-up.

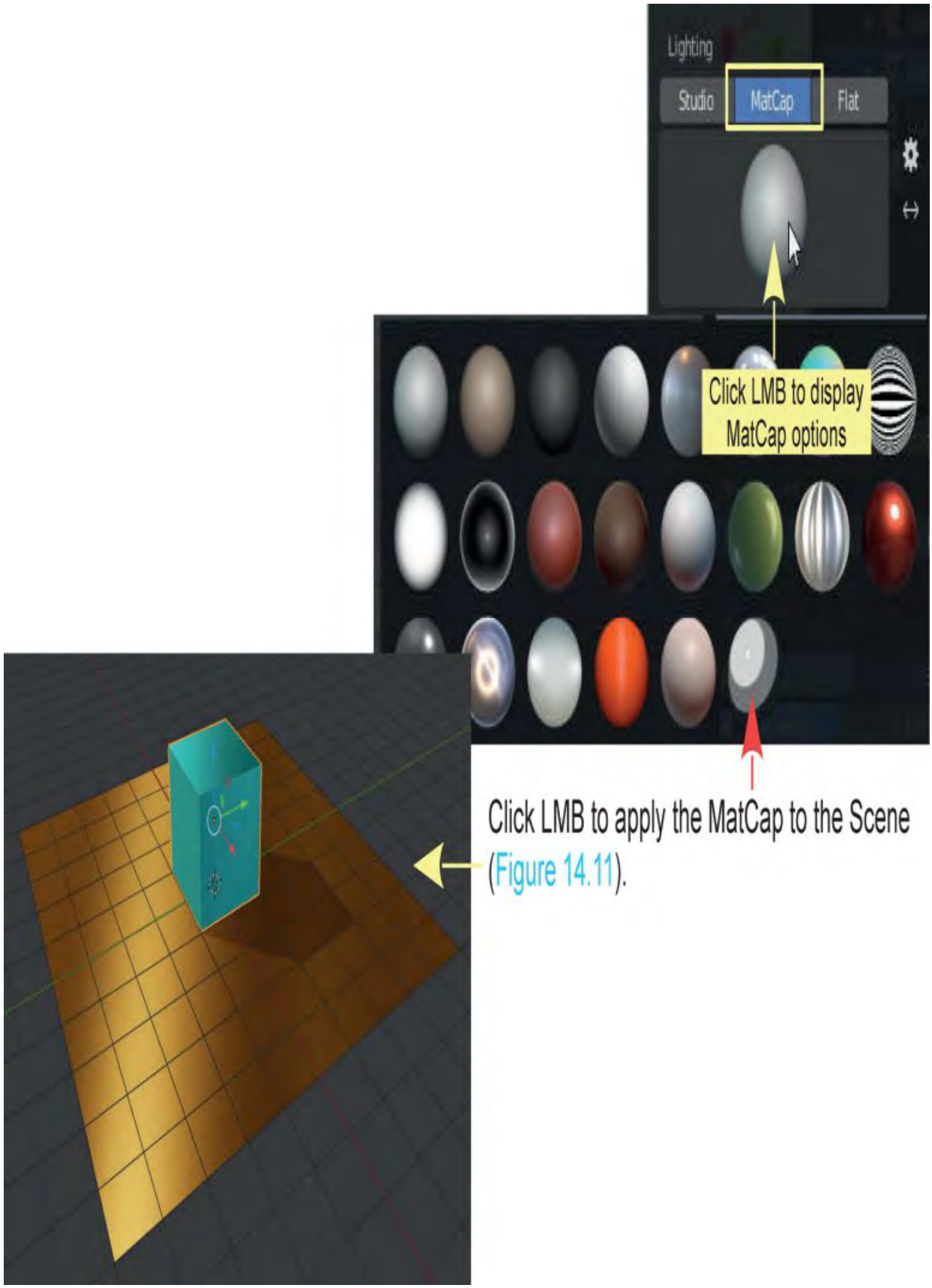

Lighting
Studio
MatCap
Flat
Click LMB to display MatCap options
Click LMB to apply the MatCap to the Scene (Figure 14.11).

Figure 14.10
Figure 14.11

With **MatCap** selected, clicking on the Sphere in the Sub Options Panel displays a selection of MatCap shading options which when pressed show you how Objects will display under different shading effects.

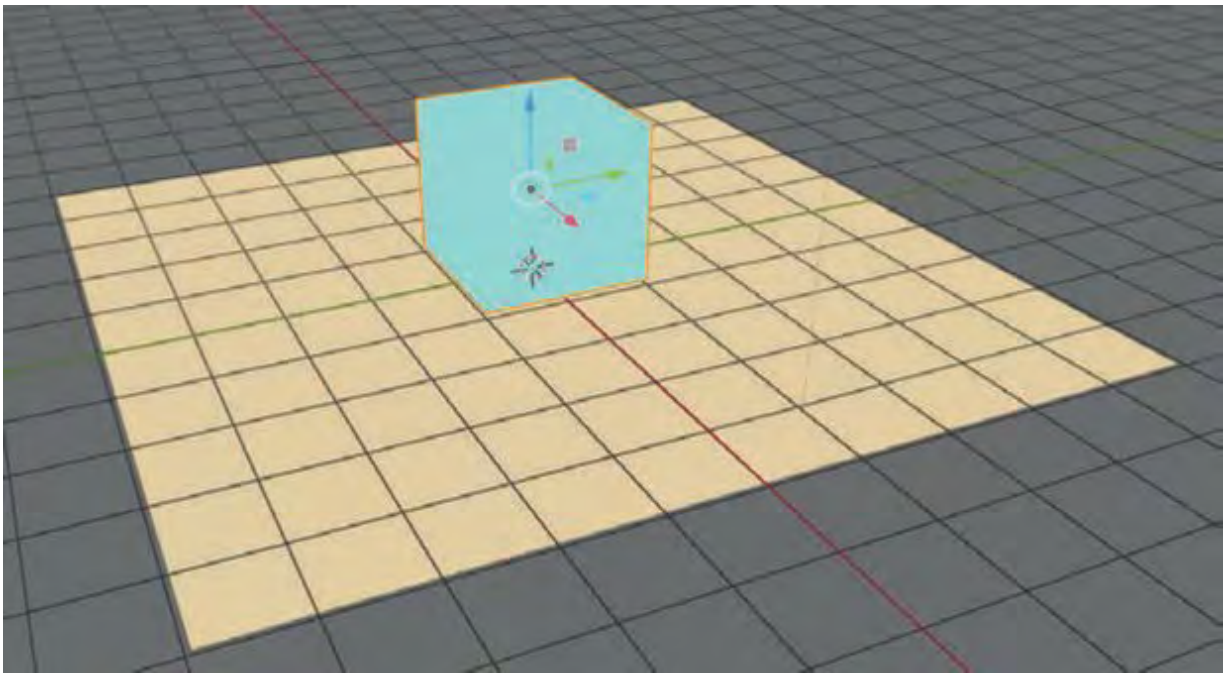


Figure 14.12

Flat Lighting applies the Material (color) set in the Properties Editor, Material buttons as a plain flat color (Figure 14.12).

Note: In the diagram, the Objects in the Scene have a Material (color) applied. The MatCap generates a lighting effect which affects the colors of the Objects giving an indication of how the Scene will appear. This effect does not render in an image.

14.4 Color Display Options (Figure 14.13)

With either **Studio Lighting, MatCap Lighting** or **Flat Lighting** there are six **Color Type** settings in the **Sub Options Panel.** The default is **Material,** which means Objects in the Scene (Viewport) are displayed with the Material color that has been assigned to the Object in the **Properties Editor, Material Properties,** when Nodes are deactivated.

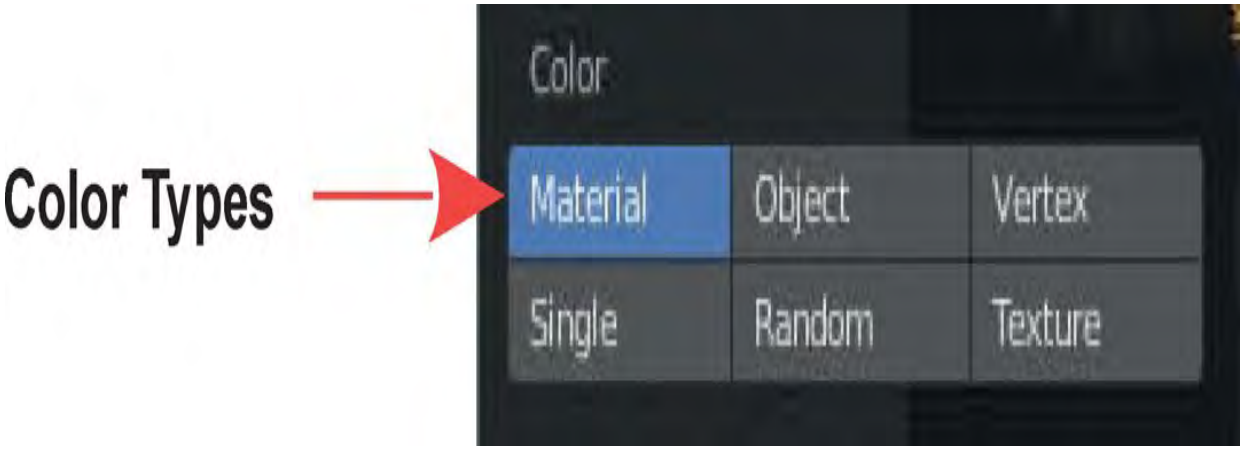


Figure 14.13

Material Color Display (Figure 14.14)

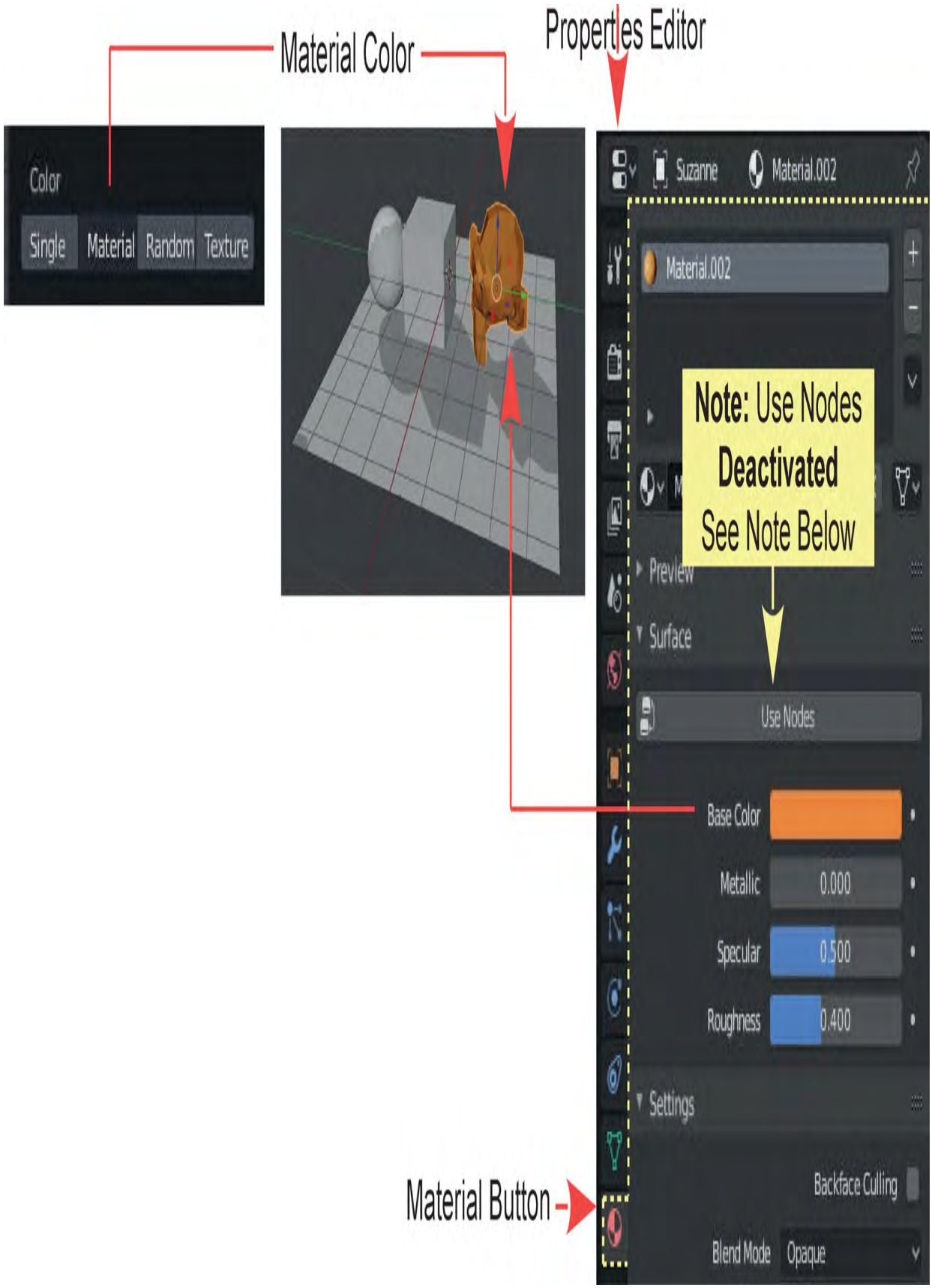

Material Color
Properties Editor
Color
Single
Material
Random
Texture
Suzanne
Material.002
Material.002
Note: Use Nodes
Deactivated
See Note Below
Preview
Surface
Use Nodes
Base Color
Metallic
0.000
Specular
0.500
Roughness
0.400
Settings
Backface Culling
Blend Mode
Opaque
Material Button

Figure 14.14

The **Base Color** applied to an Object in the Scene in the Properties Editor, Material buttons, Surface tab is displayed in the Viewport.

Note: In the diagram a Brown Material has been assigned to the Monkey Object. The Cube, Sphere and Plane Objects have the default gray Material applied.

Single Color Display (Figure 14.15)

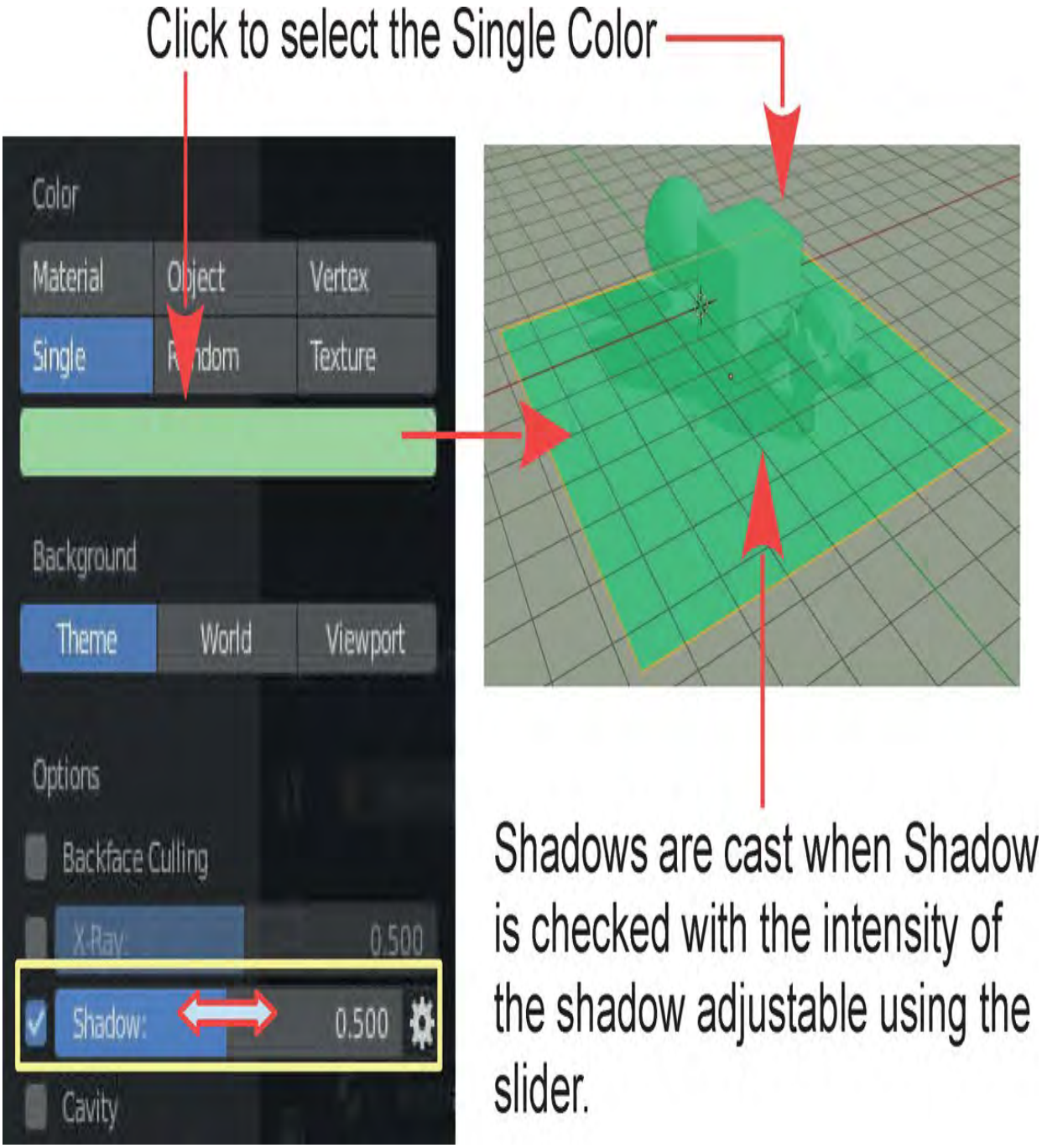


Figure 14.15

Shadows are cast when Shadow is checked with the intensity of the shadow adjustable using the slider.

Note: Material Type - When employing the **Material Type** option, the colors only show when Nodes have been **deactivated.**

Random Color Display (Figure 14.16)

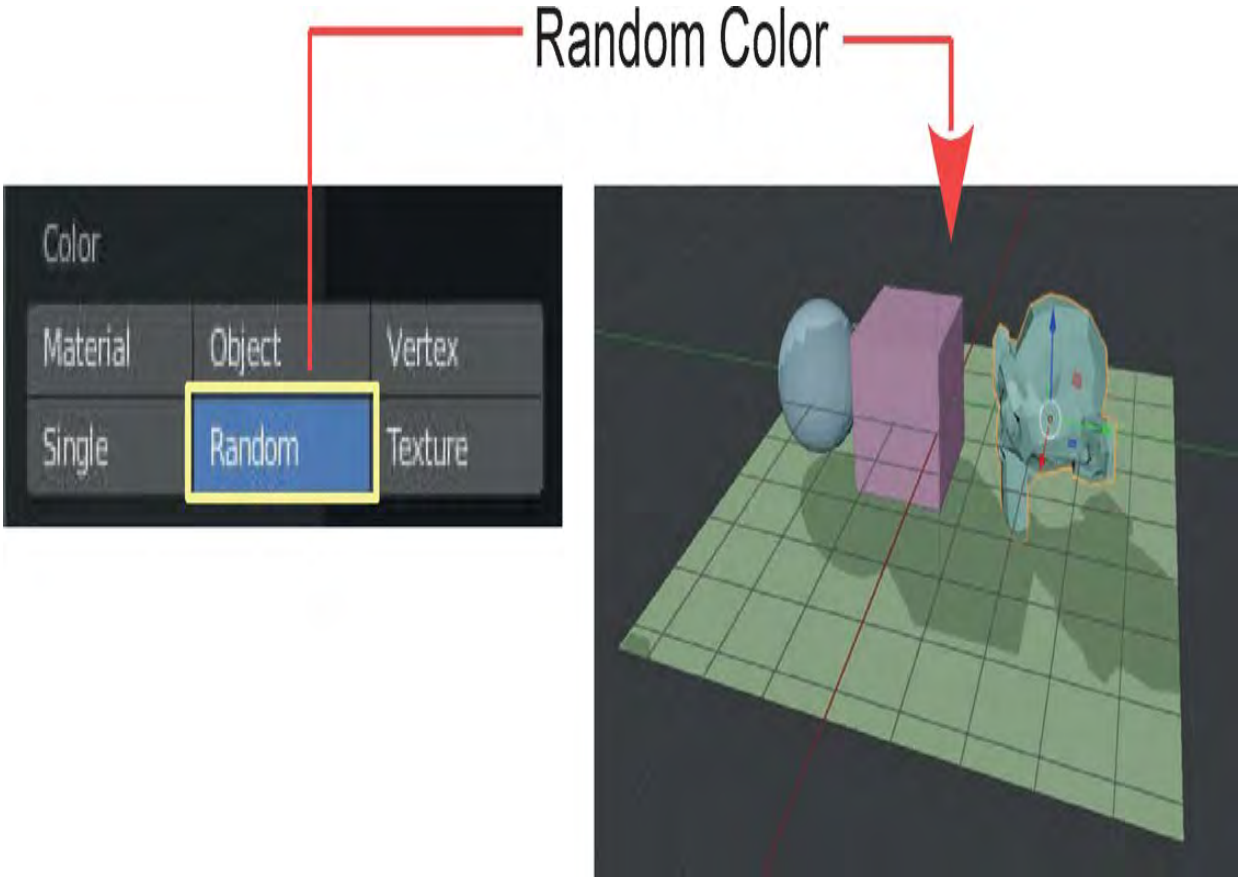


Figure 14.16

Random Color displays each Object in the Scene with a different color. The color assignment is automatic.

Texture Color Display (Figure 14.17)

Textures are effects which give the surface of an Object characteristics. Texture Color displays Objects with a Texture that has been previously set as the Base Color in the Properties Editor, Material buttons

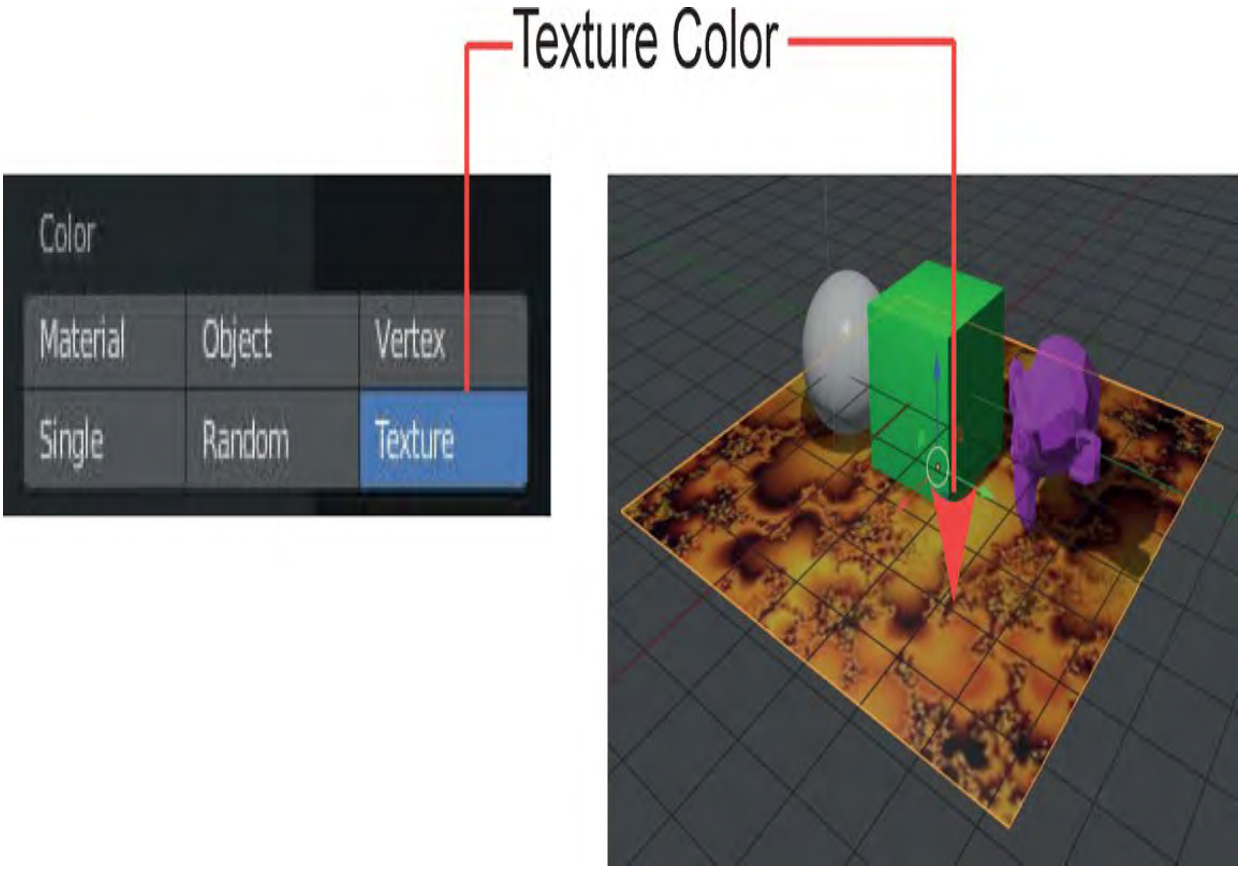


Figure 14.17

Note: You can only set the Base Color as a Texture in the Material buttons when the Node System is active.

Vertex Color Disp ay

Figure 14.18

14.5 Background Displays

In **Solid Viewport Shading Mode** there are three **Background** display options.

Figure 14.19

Theme

Background option **Theme** displays the Viewport with the Theme that has been set in the **User Preferences Editor** (Reference Chapter 2 –2.18 Figure 2.34).

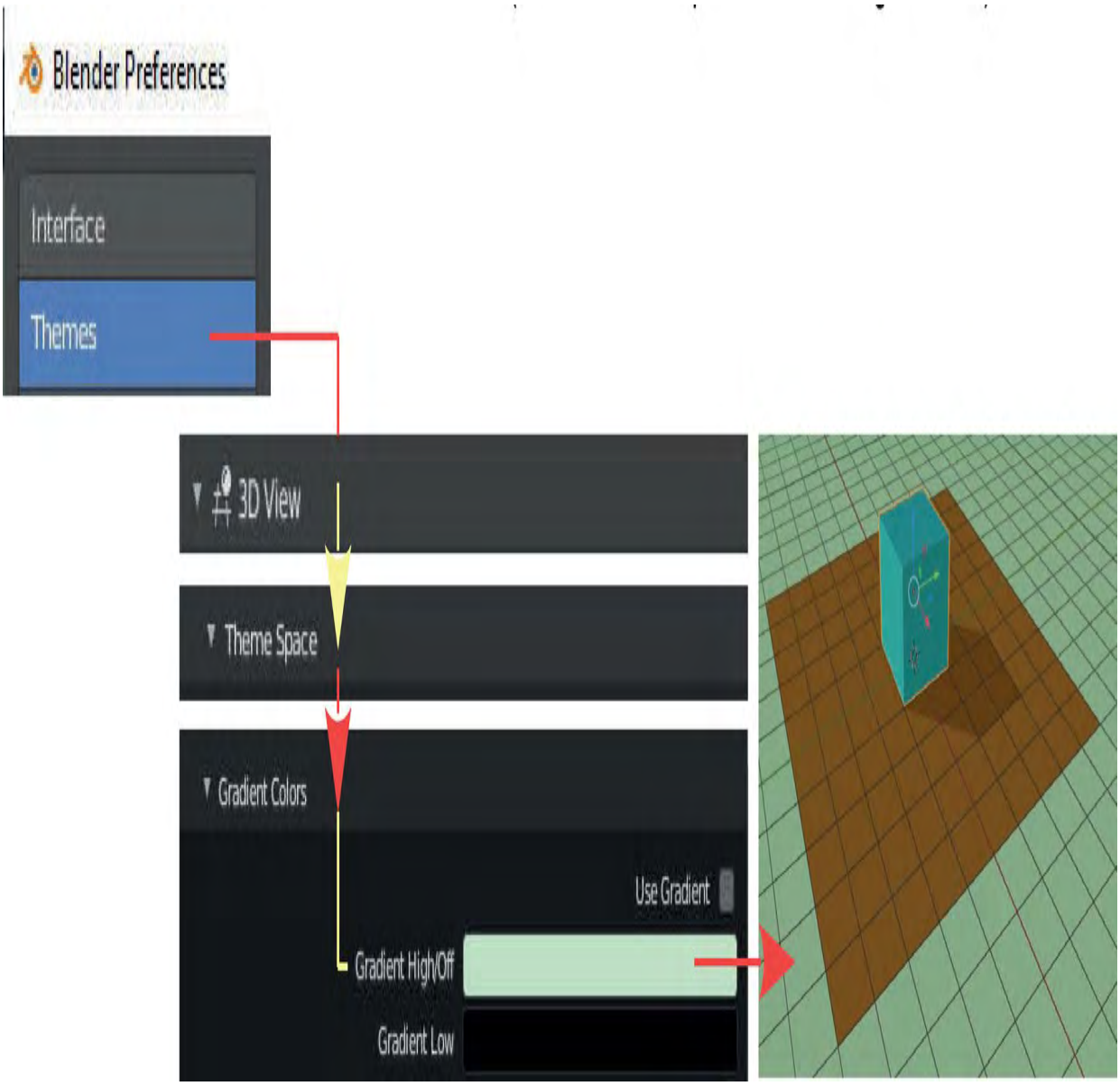


Figure 14.20

World (see 14.9 World Settings)

Background option **World** displays the Viewport with the Background that has been set in the **Properties Editor, World Properties, Surface Tab** when **Use Nodes** is deactivated (Gray).

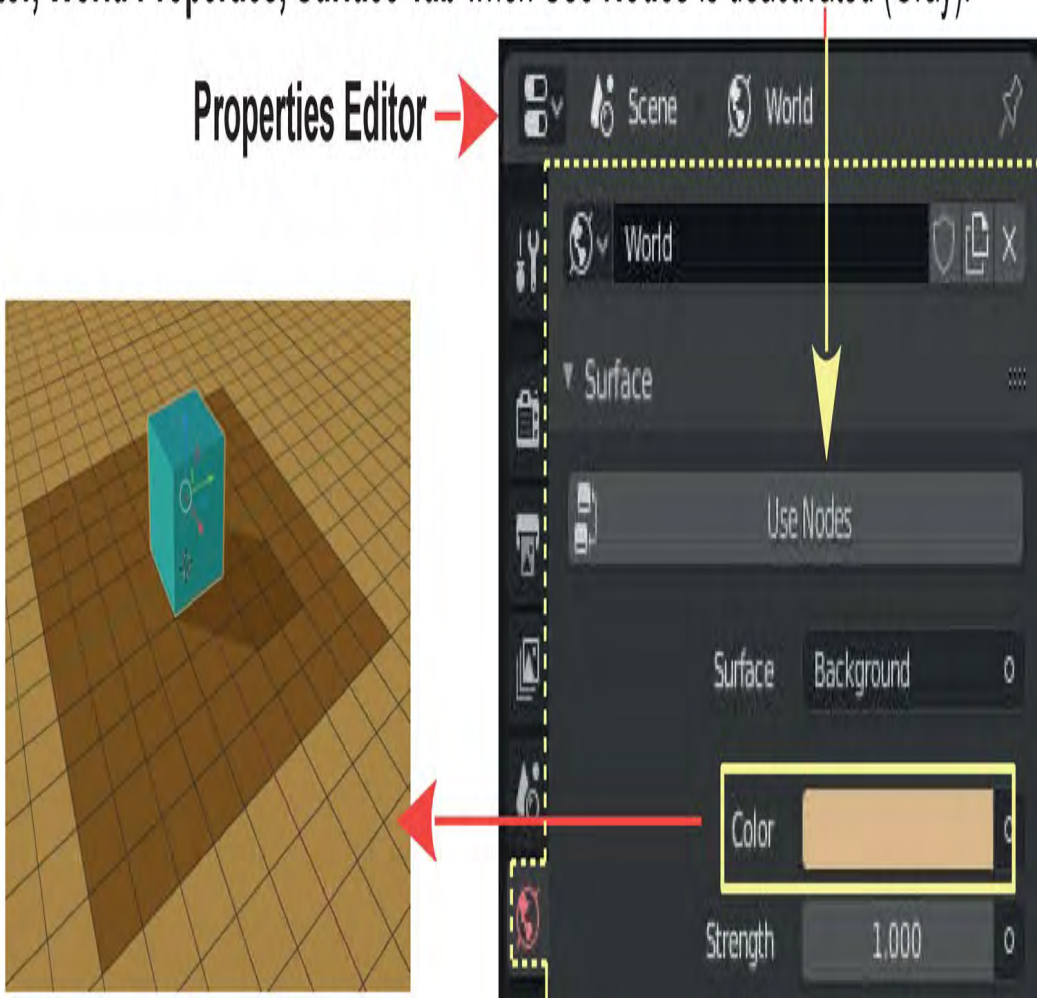


Figure 14.21

Viewport

A preview of the Scene Background may be set in the Shading Sub Options Panel.

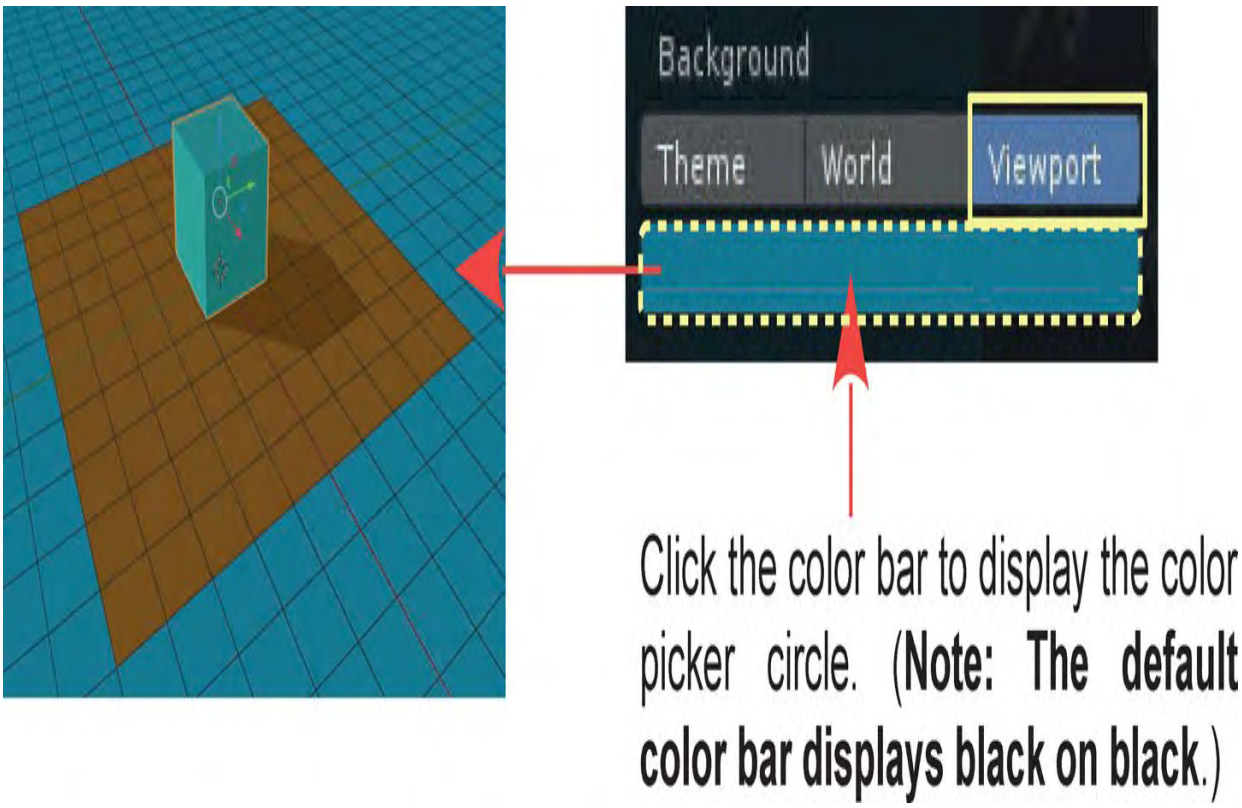
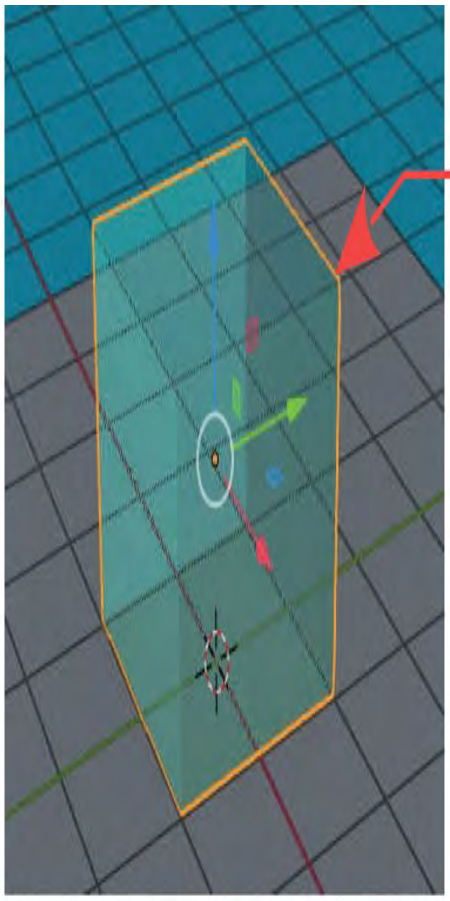
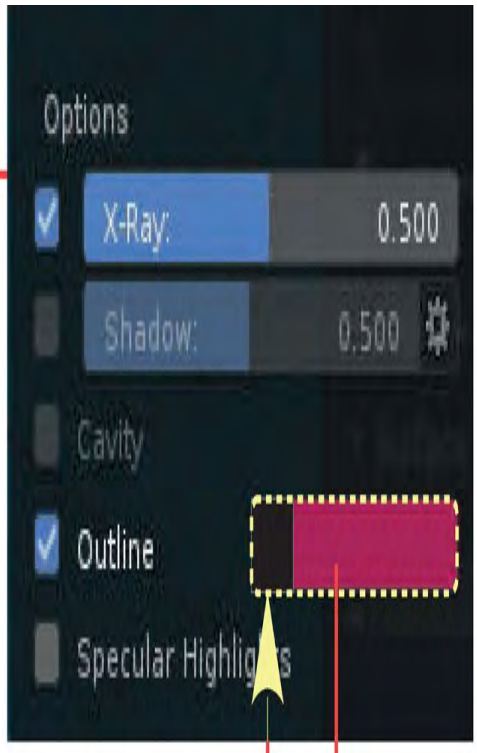


Figure 14.22

14.6 More Solid Viewport Shading





Color bar displays Black on Black

X-Ray view of the selected Object

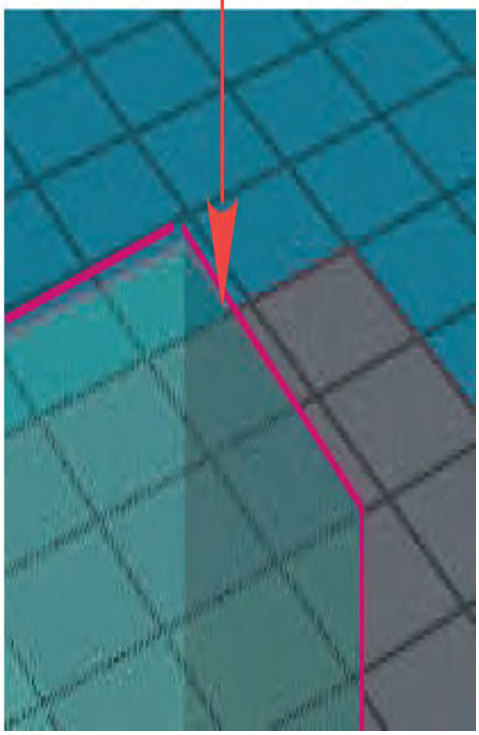


Figure 14.23
Figure 14.24

Shadows: Previously discussed.

Outline: Object Outlines are displayed orange when selected. When **deselected** they display with a color which you choose by clicking the color bar next to **Outline** in the Sub Options panel. The colored outline may be turned turned off by unchecking (unticking) **Outline** in the panel.

14.7 Rendered Viewport Shading

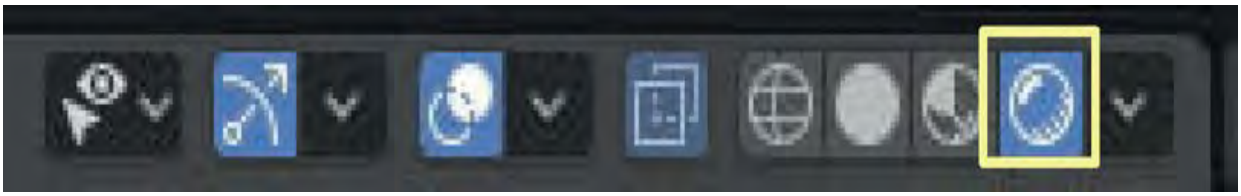


Figure 14.25

Rendered Viewport Shading gives a quick access to a preview showing exactly what you will see in a Rendered view. Unlike Solid and Material Preview Viewport Shading Modes it has no sub options. Clicking Rendered Viewport Shading displays the 3D Viewport Editor as a Rendered view and as such, the Lighting in the view is influenced by the Lamps and lighting arrangement set in the Scene.

Note: Textures used to color objects will NOT display in a Rendered view unless the Material **Node System** has been employed.

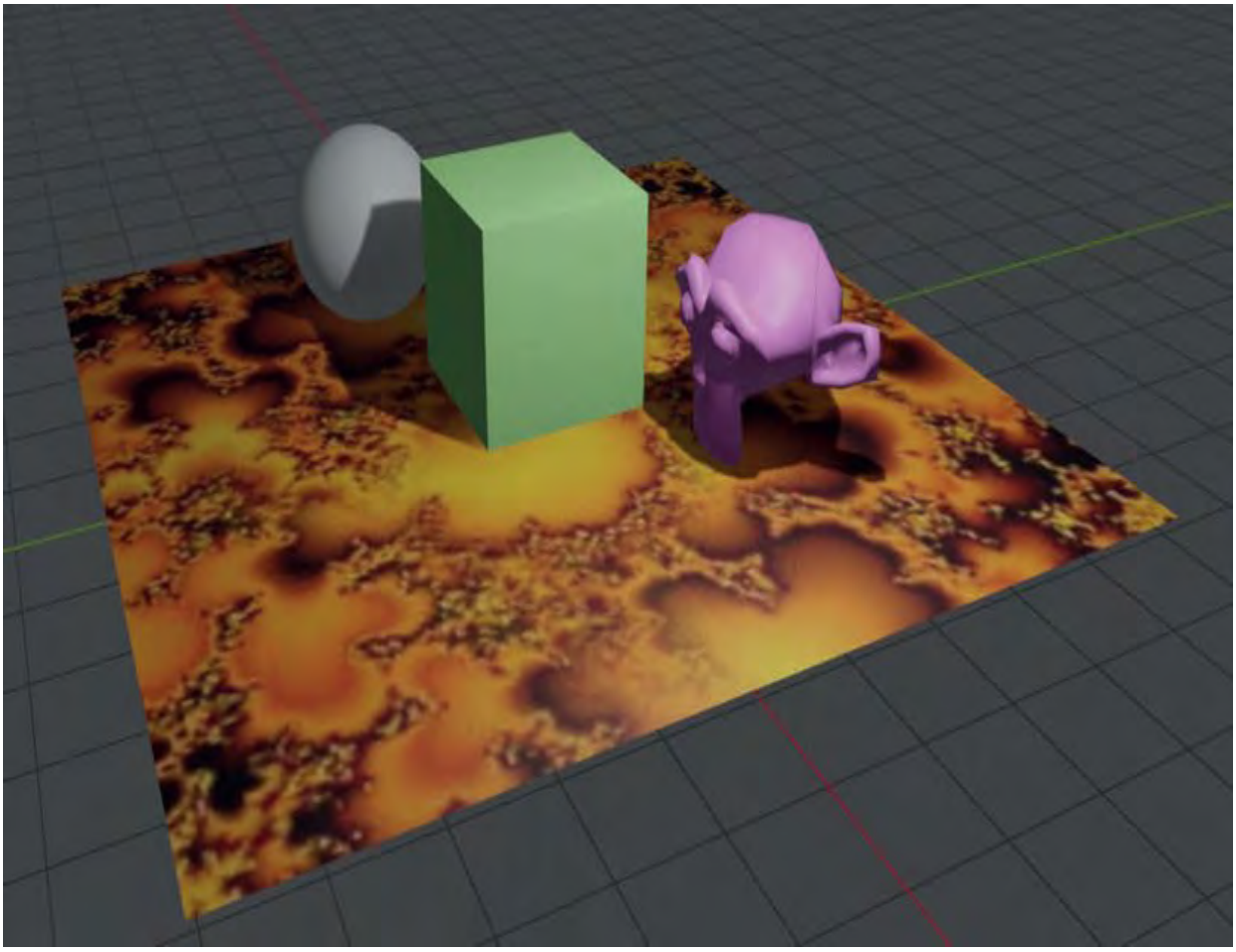


Figure 14.26

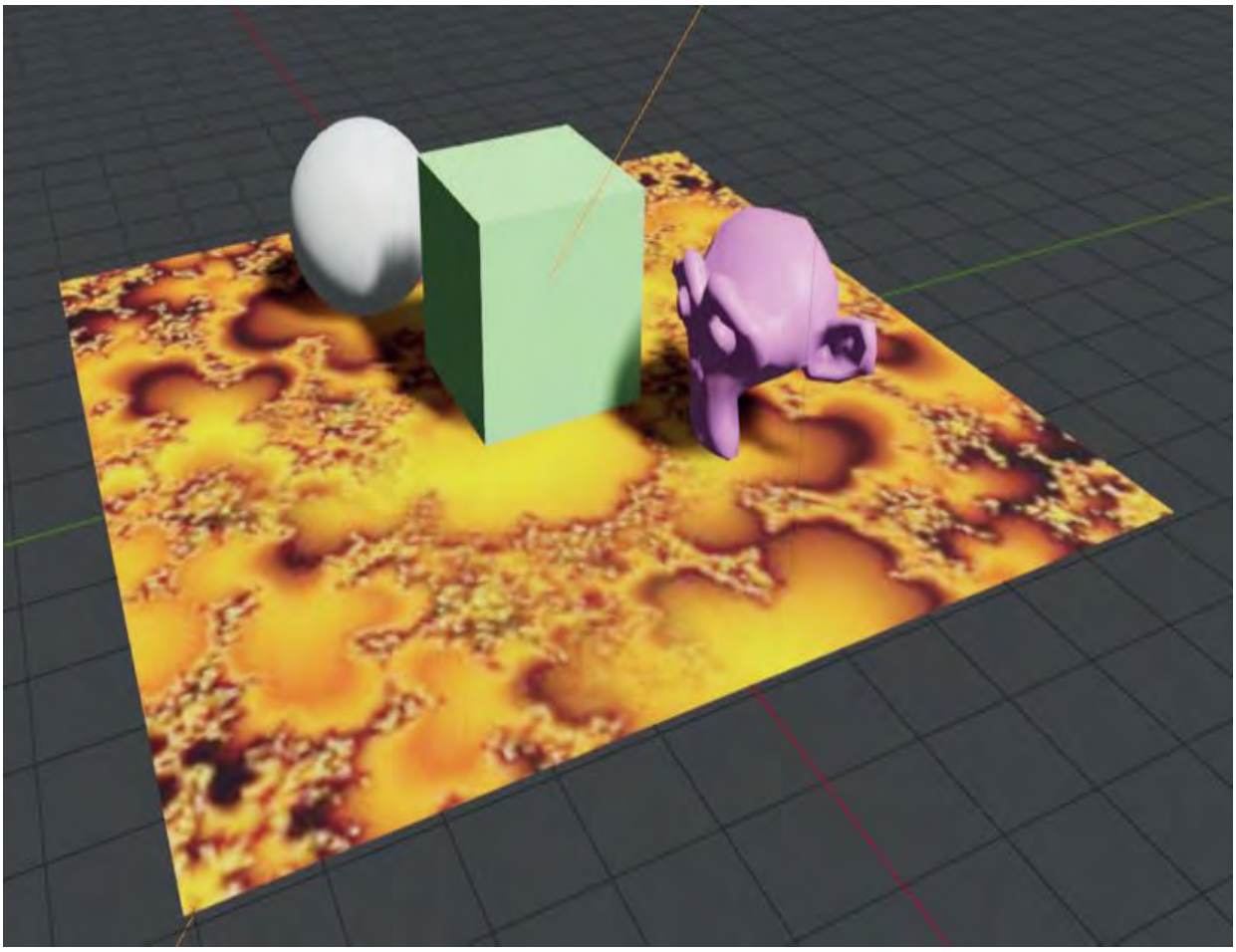


Figure 14.27

In Figures 14.26 and 14.27 the Objects have Materials (colors) applied using the Material Node System (Reference: Chapter 16) in the Properties Editor, Materials buttons (Figure 14.28). The Materials Shader Editor has NOT been used, instead, with Use Nodes active in the Properties Editor Base Colors have been selected by clicking on the Base Color bar.

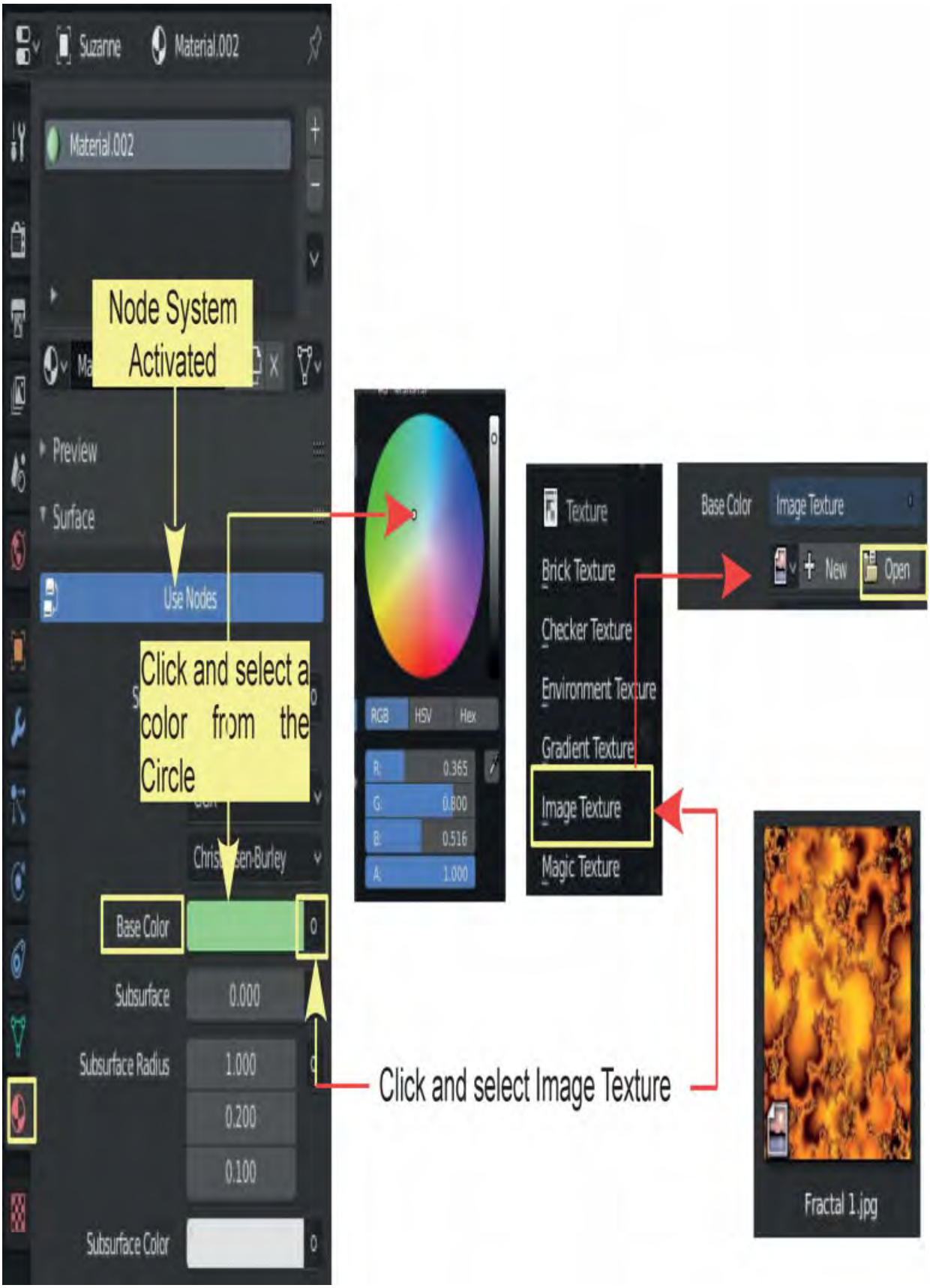

Suzanne
Material.002
Material.002
Node System Activated
Preview
Surface
Use Nodes
Click and select a color from the Circle
Base Color
Subsurface
0.000
Subsurface Radius
1.000
0.200
0.100
Subsurface Color
RGB
HSV
Hex
R: 0.365
G: 0.800
B: 0.516
A: 1.000
Texture
Brick Texture
Checker Texture
Environment Texture
Gradient Texture
Image Texture
Magic Texture
Base Color
Image Texture
New
Open
Click and select Image Texture
Fractal 1.jpg

Figure 14.28

Click Open, navigate and select an Image

In the case of the Plane Object an Image Texture has been used for the color.

14.8 Material Preview Shading

Figure 14.29

When creating or modifying a Scene using Solid Viewport Shading you may wish to quickly view the Scene as it appears with illumination provided by the Lamps instead of the illumination from the Solid Shading Mode. At times illumination is provided by special HDRI images used as a background to the Scene (Reference Chapter 15 – 15.9). If an HDRI image has been installed and you are in Solid Viewport Shading Mode Material Preview provides a quick preview.

Figure 14.30 shows the Scene in the 3D Viewport Editor in Solid Viewport Shading Mode with the Monkey Object selected in Edit Mode. If this were a complicated Scene where the Monkey was obscured by shadow you could clearly see the Monkey and Vertices.

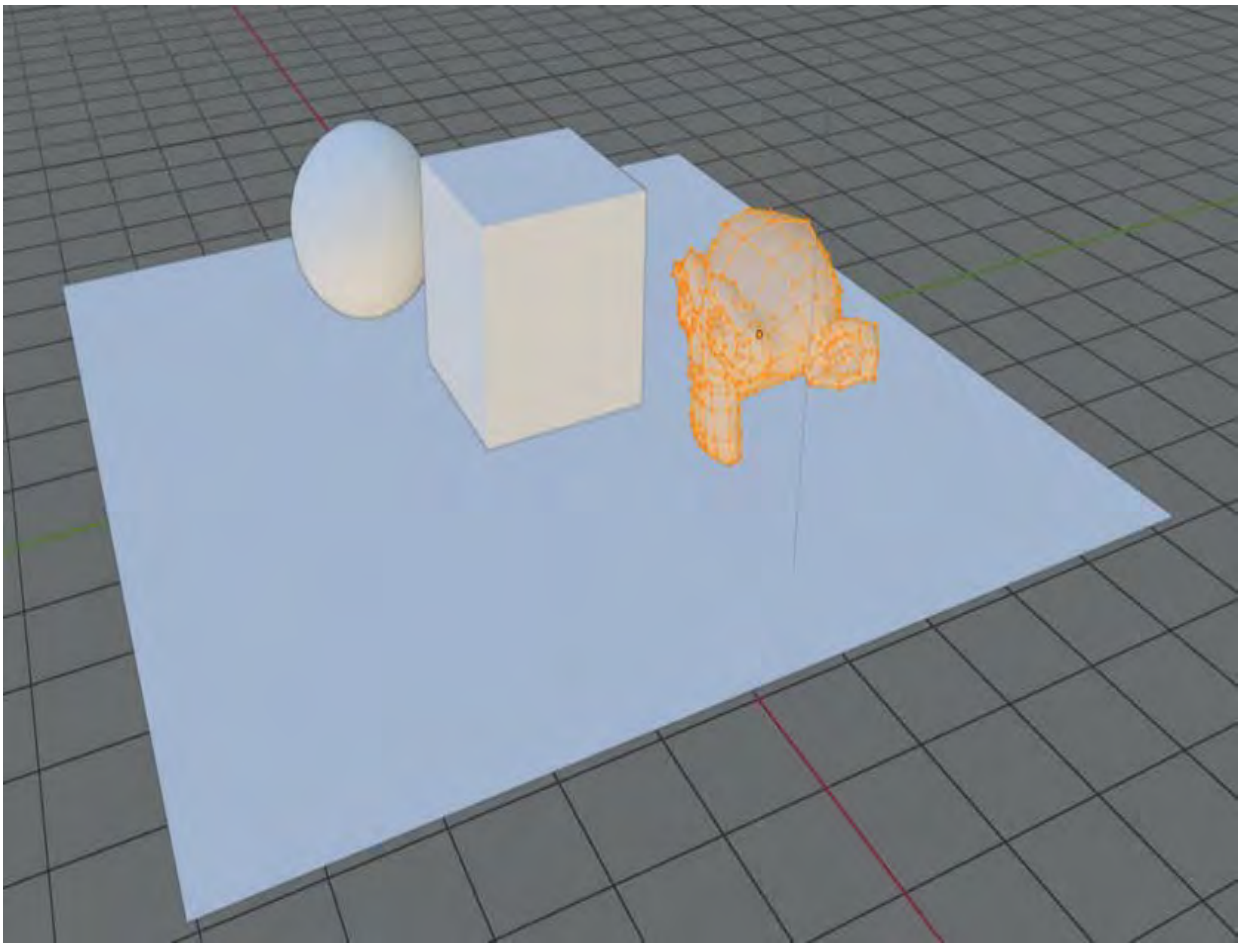


Figure 14.30

The Scene is viewed under Studio Lighting.

To quickly see how the Scene looks with the edits to the Monkey select **Material Preview Shading** (Figure 14.31).

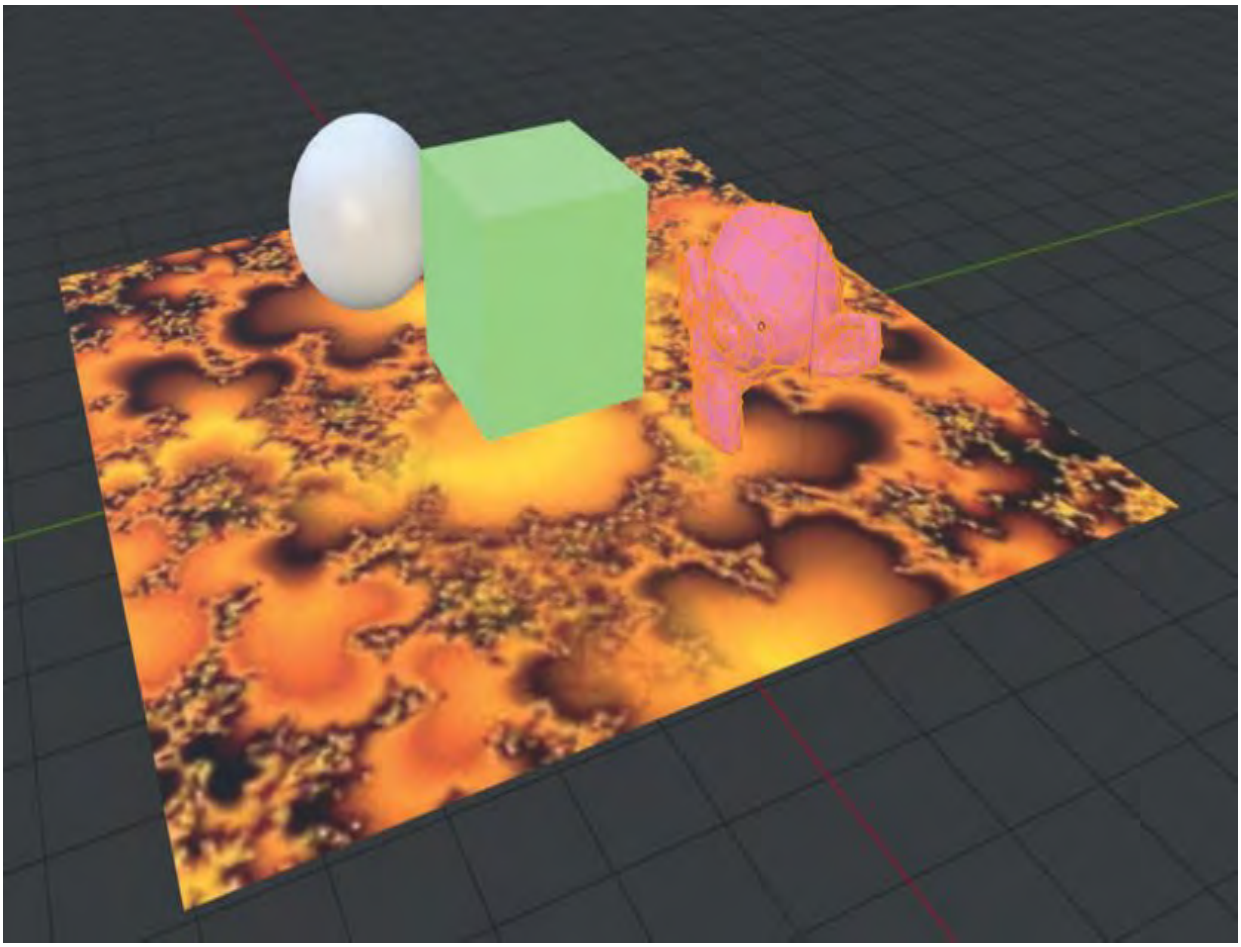


Figure 14.31

In the Sub Options select **Scene Lighting** to view the Scene using the lighting effects in the Scene (Lamps)

Lighting
Scene Lights
Scene World
Rotation:
0°
Background:
0.000

rigure 14.32

You may use the **Matcap** to preview how adjustments to Lamp setting would affect the display.

If an image has been used to illuminate the Scene select **Scene World** instead of Scene Lights to preview the Scene (Figure 14.32)

HDRI Image used as Scene background (Reference Chapter 15– 15.9).

14.9 World Settings

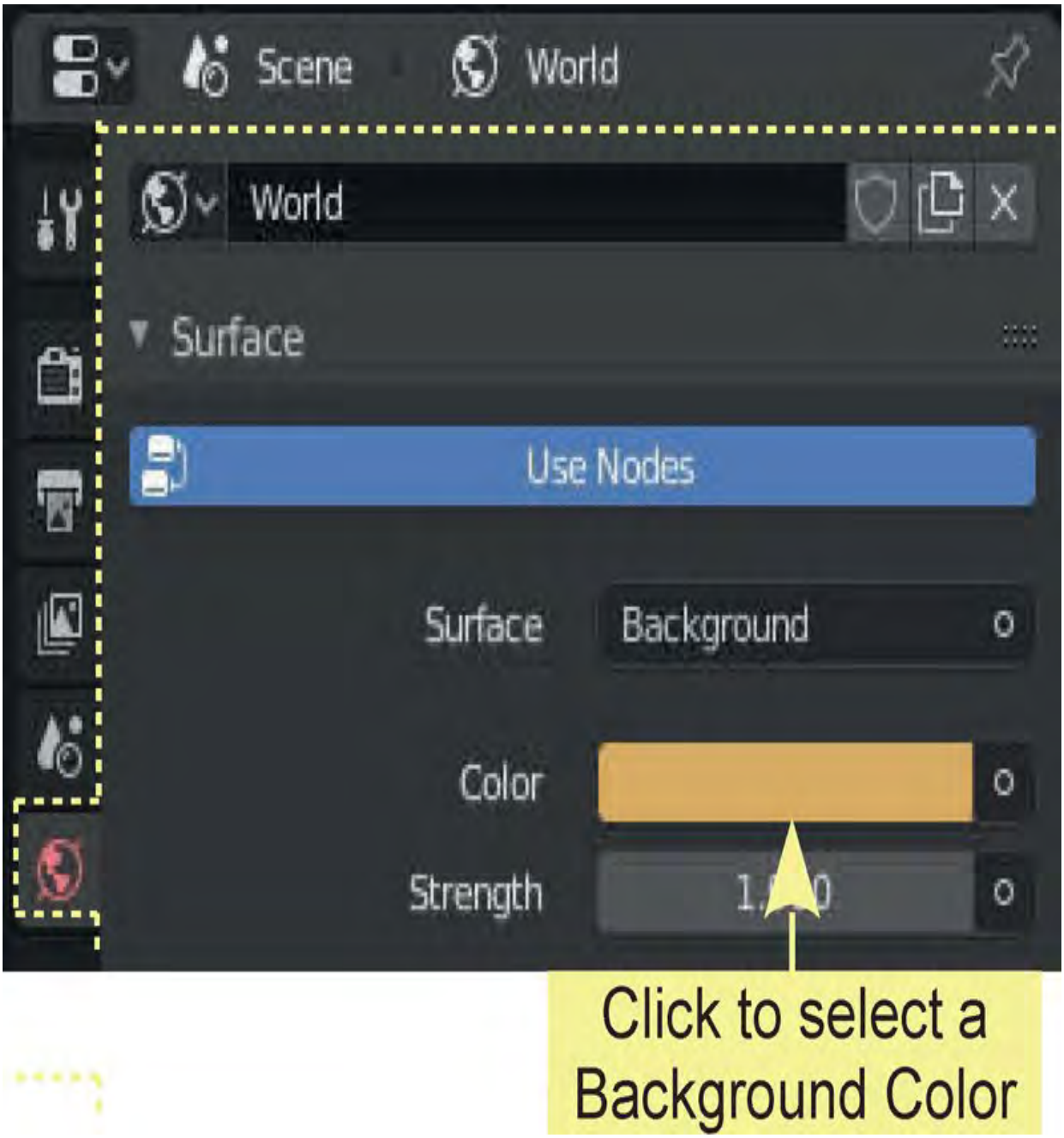


Figure 14.33

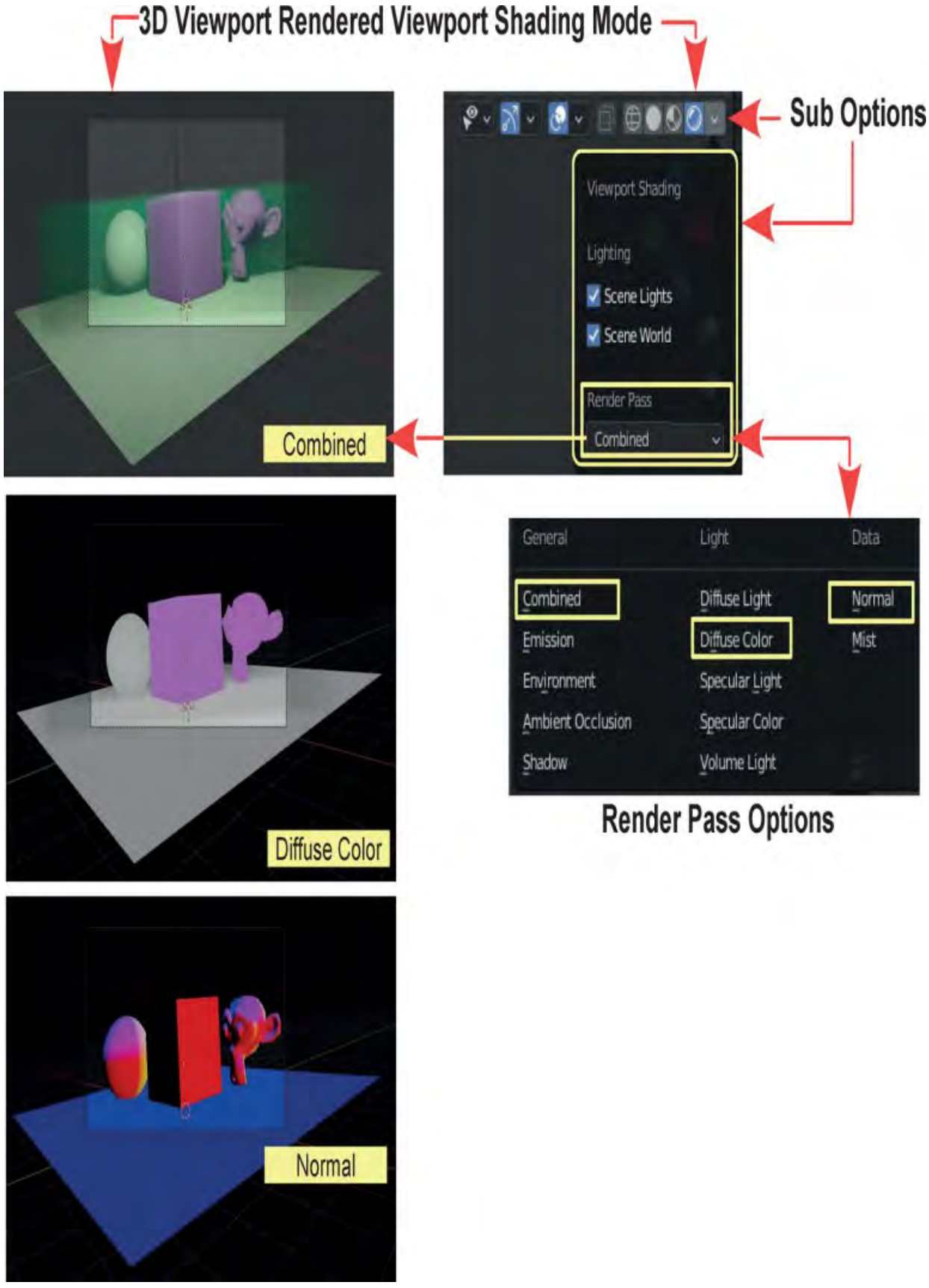
World background color settings in the Properties Editor, World Properties also influence the display in the 3D Viewport Editor (Figure 14.33).

Note: The background color set in the World Properties can be seen in the 3D Viewport Editor in Rendered Viewport Shading Mode or in Solid

Viewport Shading with Background World activated or in Material Preview with Scene World checked.

14.10 Simplified Viewport Shading

At the beginning of the chapter two images depicted Rendered Viewport Shading and Simplified Viewport Shading. **Simplified Shading** is a reference to a selection of Display Modes called **Render Passes.** Simply put, when a Scene is Rendered, Blender performs the operation in a series of passes which when combined produces the Rendered View. The passes may be isolated and viewed in the 3D Viewport Editor allowing adjustments to be made.



Render Pass Options

Figure 14.34

In Figure 14.34, **Combined** displays the Combined Passes, **Diffuse Color** displays the Diffuse Color of Objects in the Scene and **Normal** colors Objects indicating the positive and negative direction of Normals (reference Chapter 24 – 24.9).

15 Scene Lighting & Cameras

15.1 Scene Lighting
15.2 Light Types
15.3 Cameras
15.4 Camera Settings
15.5 Camera Switching
15.6 Camera Tracking
15.7 Basic Lighting Arrangement
15.8 Background Scene Lighting
15.9 Images as a Background
15.10 Volumetric Lighting

Introduction

The **Camera** or cameras in the 3D Viewport Editor capture the part of the Scene that is to be Rendered as a still Image File or Movie File. That part of the Scene is termed **Camera View. Scene Lighting** creates the mood of a Scene determining whether it depicts a bright happy sunny event or something that is dark and sinister. Lighting is, therefore, a significant factor in creating a still Image or creating atmosphere in a Video Clip or Movie.

lt is important to understand that Scene Lighting only has effect when the 3D Viewport Editor is in **Rendered Viewport Shading Mode.**

The default Blender Scene is displayed in **Solid Viewport Shading Mode** with **Lighting: Studio** and **Color: Material** set in the Shading Options Panel (see Chapter 14 - 14.3). Solid Viewport Shading is primarily used for creating models and setting the physical aspects of the Scene. The single Light

in the default Scene has no effect in the 3D Viewport Editor when **in Solid Viewport Shading Mode.** The default Scene is illuminated by an arbitrary light source which is perceived to be above the mid-plane grid to the left of the default Scene displayed in User Perspective View.

The **Light** in the default Scene has effect when the 3D Viewport Editor is in Rendered Viewport Shading Mode which represents what will be Rendered from Camera View.

15.1 Scene Lighting

The default Scene in the 3D Viewport Editor is illuminated by **Studio Lighting** when in **Solid Viewport Shading Mode** (Reference Chapter 14 - 14.3). This Shading Mode is independent of **Light Objects,** therefore, the default Point Light in the default Blender Scene has no affect. To understand Blender Lights, change the Viewport Shading to **Material Preview Viewport Shading** and activate **Scene Lights** and **Scene World** in the **Sub Options Panel.**

Figure 15.1

15.2 Light Type

The default Blender Scene contains a single **Point Light Object.**

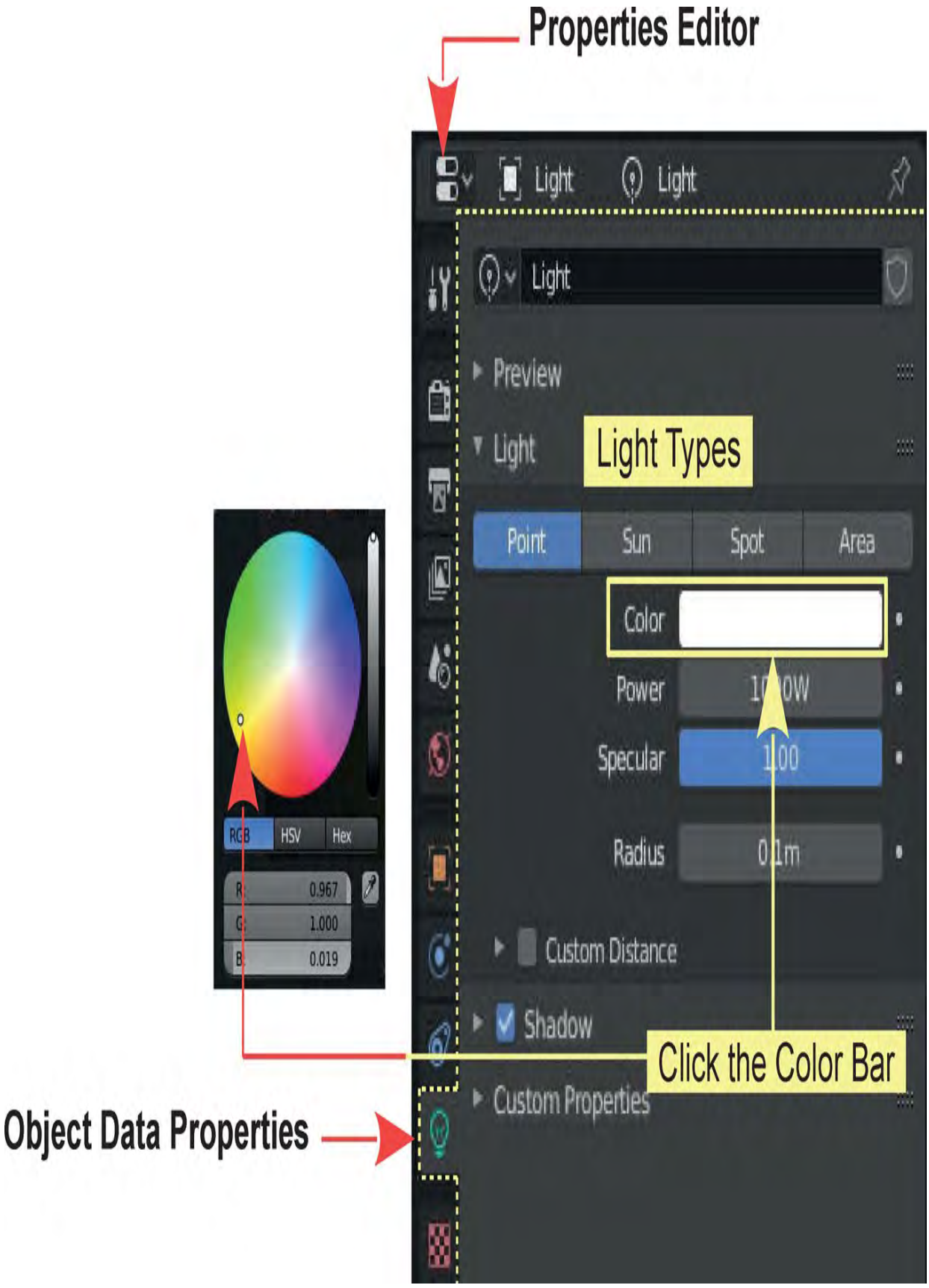

Properties Editor
Light
Light
Light
Preview
Light
Light Types
Point
Sun
Spot
Area
Color
Power
Specular
Radius
Custom Distance
Shadow
Custom Properties
RGB
HSV
Hex
0.967
1.000
0.019
Click the Color Bar
Object Data Properties

Figure 15.2

With the **Light** selected in the 3D Viewport Editor, go to the **Properties Editor, Object Data Properties** to display the setting options (Figure 15.2). You may change the Light by selecting one of the types in the **Light Tab.**

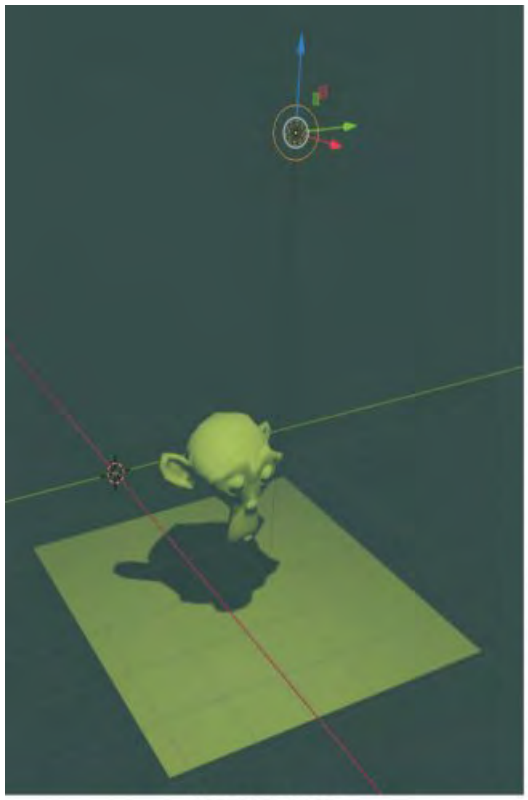
The Properties Editor display will vary depending on the Light Type selected.

The color of light may be selected by clicking the color bar to display a color picker circle.

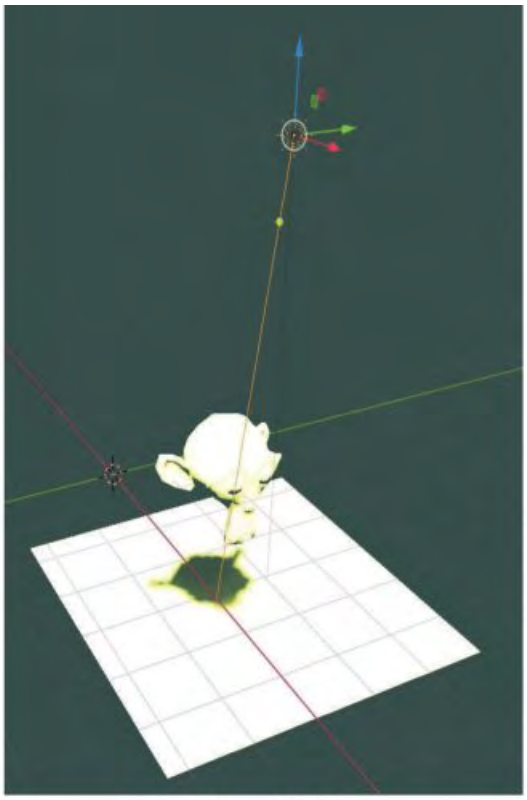
Note: The Sun Light is not affected by Color Change.

Lights in **Blender** are considered to be Objects and as such they may be Translated and Rotated like other Object. Each Light Type has properties which may be adjusted in the Properties Editor when the Light is selected in the 3D Viewport Editor.

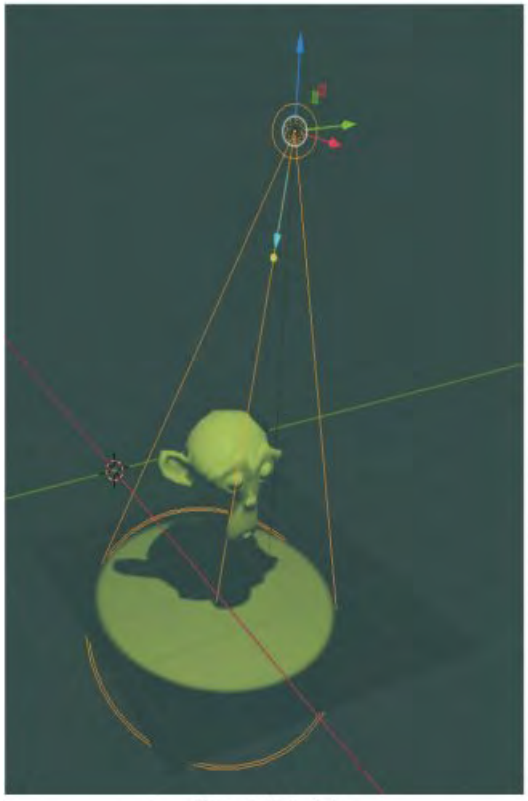
The diagrammatic representation of the Light Type changes in the 3D Viewport Editor depending on the Type selected (Figure 15.3).



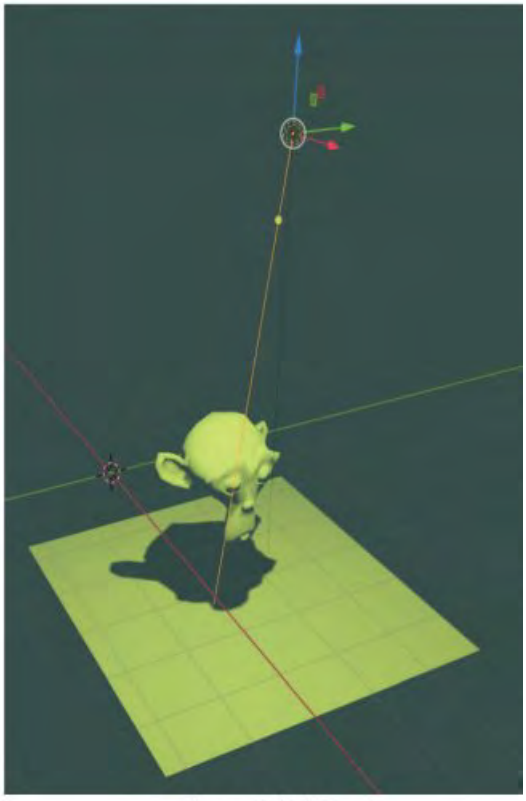
Point Light



Sun Light



Spot Light



Area Light

Figure 15.3

Remember: The 3D Viewport is in **Material Preview Viewport Shading Mode** with Scene Lights and Scene World checked in the Options. In Figure 15.3 the different Light Types illuminate the Scene. In each example the Light Color, set in the Properties Editor, Object Data Properties, with the Light Object in the 3D Viewport Editor selected, casts the color (green) on to the Monkey and the Plane. The Monkey and the Plane do not have a Material (color) applied and, therefore, would display with the default gray without the Light Color being cast by the Light. In conjunction with the illumination from the Lights the World Surface Background settings in the Properties Editor, World Properties are influencing the way in which the Scene displays (Figure 15.4).

Properties Editor

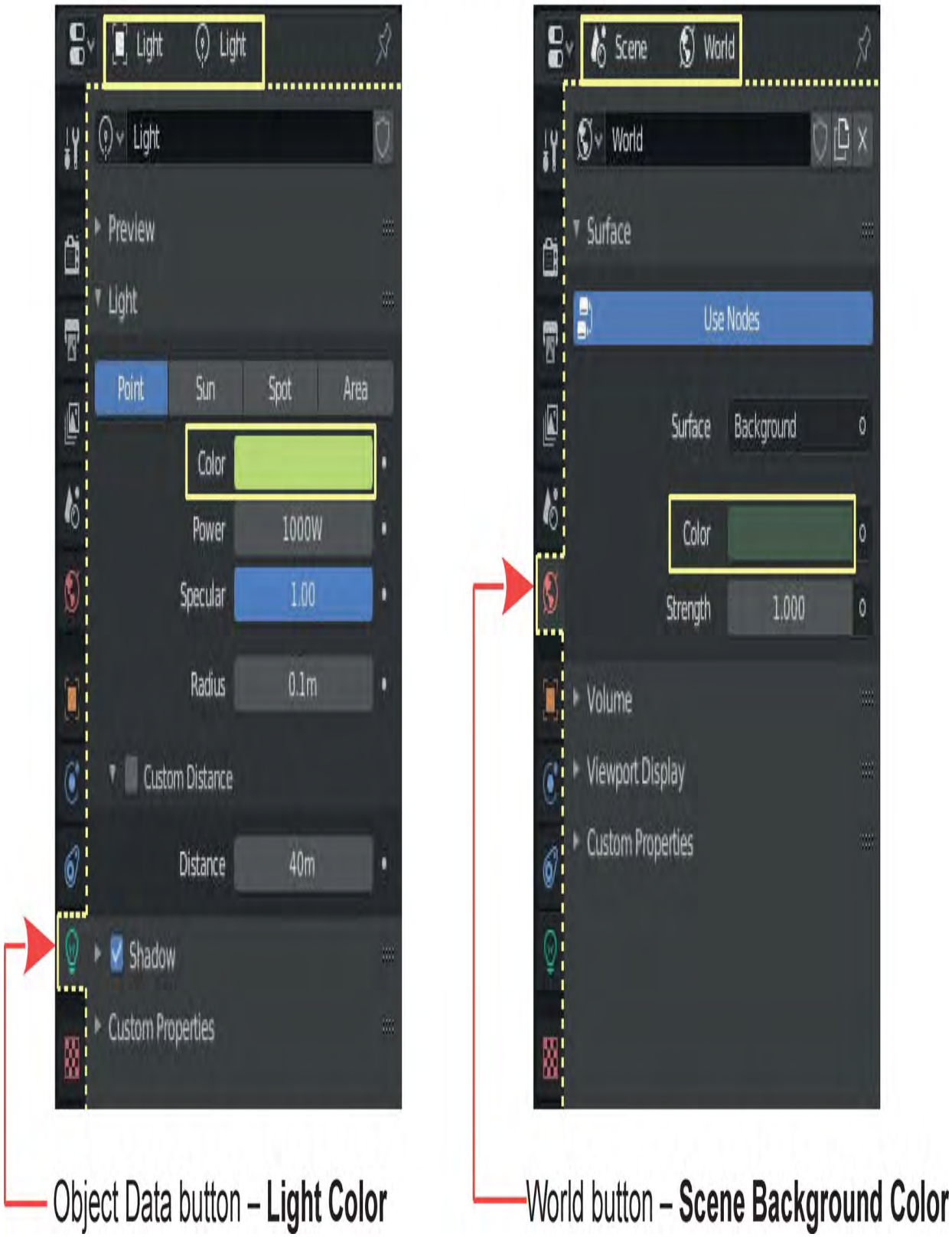


Figure 15.4

Adding Lights

To see the effect of the different Light Types and settings create a Scene as shown in Figure 15.5.

Scene Arrangement

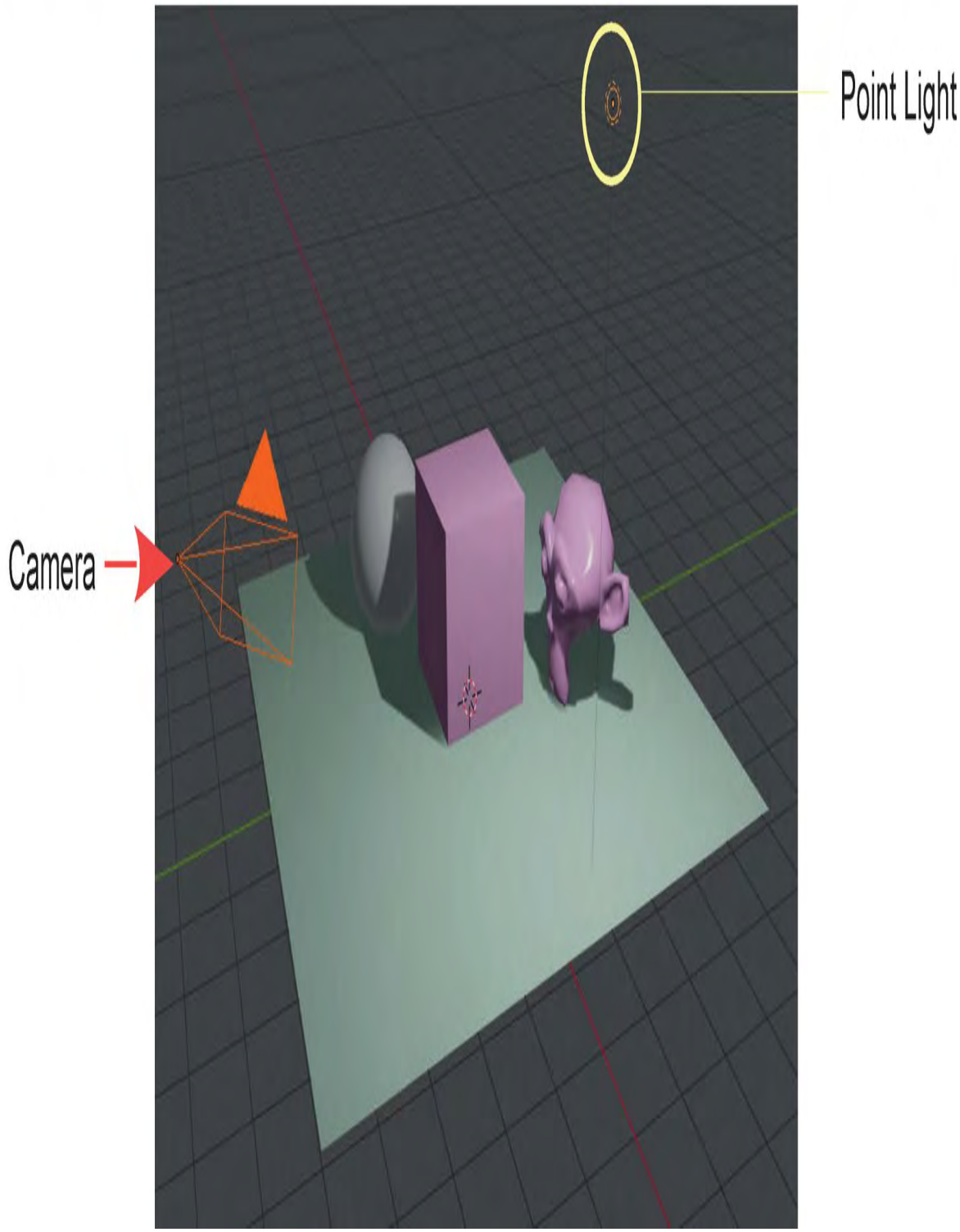


Figure 15.5

In the Scene a Plane has been added and Scaled up five times. The Plane has been Translated (Moved) up, just above the Midplane Grid. A UV Sphere and a Monkey have been added and positioned each side of the default Cube. The Cube, UV Sphere and Monkey have been Translated up to sit just above the Plane.

Materials

Pale Magenta Material Color has been added to the Cube and Monkey with the Node System active (Reference: Chapter 4 – 4.18). Material has NOT been added to the UV Sphere or the Plane.

The Camera and the default Point Light are in their default positions. The 3D Viewport Editor is in Material Preview, Viewport Shading Mode (Chapter 14 – 14.4) with Scene lights and Scene World checked in the Sub

Lighting Effects

The default Point Light has a pale green color applied.

Make note of the different effect when you change the Light Type in the Properties Editor, Object Data Properties, Light tab (with the Lamp selected).

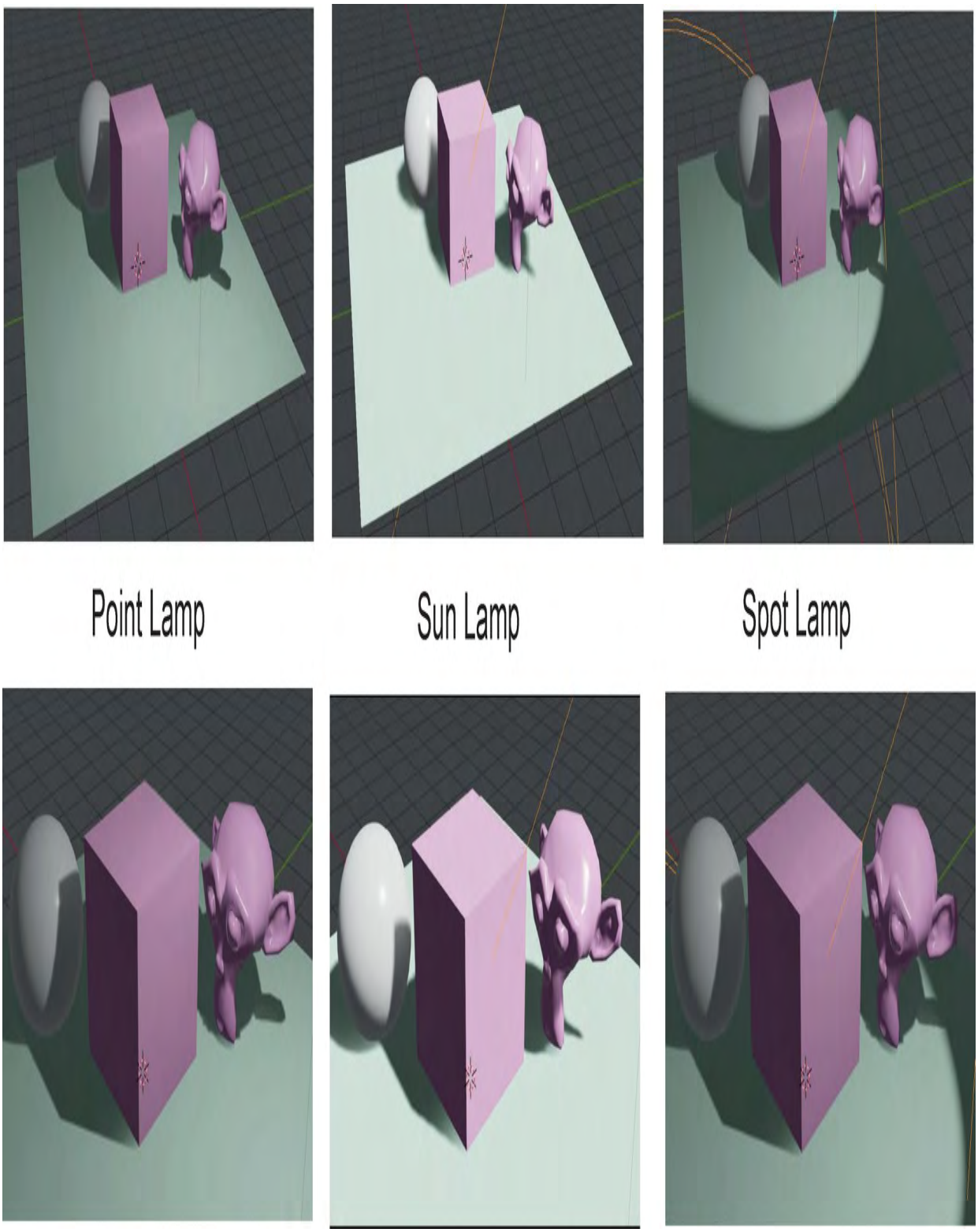


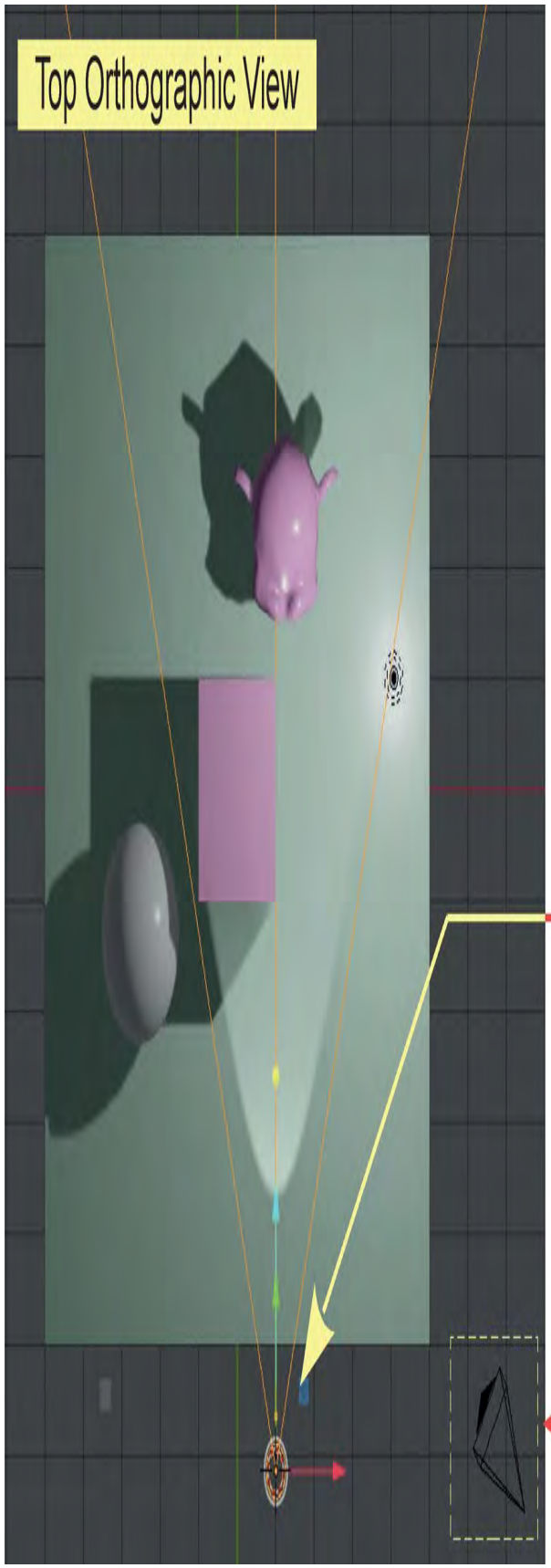
Figure 15.6

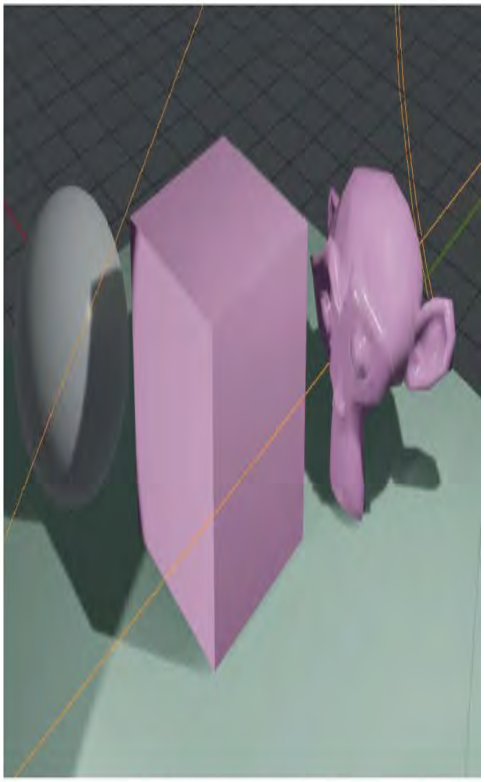
The upper row of images show part of the 3D Viewport Editor. The lower row shows Camera View. Although a Material Color has NOT been applied to the UV Sphere or the Plane they display with the pale green color cast by the Light. The Cube and Monkey show the Pale Magenta Material although it is tempered by the Light color. **Note:** Your display will vary depending on the colors that you have applied.

By leaving the default Point Light in position, then adding an additional Spot Light and directing it towards the left hand face of the Cube you remove the shadow on the Cube and the Monkey (Figure 15.6).

To add additional Lights to the Scene, position the cursor in the 3D Viewport Editor, press **Shift + A Key** and select **Light** from the menu that displays. You can choose **Point, Sun, Spot or Area.**

Additional Light Added





Camera View

Additional Spot Light pointing towards the Cube and the Monkey.

The Camera in its default position.

Figure 15.7

15.3 Cameras

By default the Scene has one **Camera** which is positioned to capture an image of the Cube Object. What the Camera sees and what is captured as an image is called **Camera View.** You can see Camera View in the 3D Viewport Editor by pressing **Num Pad O** on the Keyboard. To return to User Perspective View, press Num Pad 5 (User Orthographic View) then Num Pad 5 again for User Perspective View. **You have to rotate the View to reinstate the default Scene or press Num Pad 6.**

In a complex Scene you may wish to add more Cameras to capture shots from different angles. You add Cameras by pressing **Shift + A key** and selecting **Camera** from the menu or click Add in the Header and select Camera. The new Camera will be located where the 3D Viewport Cursor is positioned. If you add a Camera in the default Scene with the 3D Viewport Cursor at the center of the Scene it will coincide with the default Cube Object. Click the Move Tool in the Tool Panel and move the Camera to one side. You have to rotate the new Camera to capture the part of the Scene you require.

Depending on the Scene arrangement the Cameras may or may not be visible in the 3D Viewport. Camera View depends on which Camera is selected. If the Camera is visible LMB click to select. If it is not visible you can select it in the Outliner Editor by clicking LMB on the name (See Notes following). The original Camera is named **Camera** the new Camera is named **Camera.001.** You may rename these to something meaningful if you wish.

Note: If you select **Camera** (the original Camera) and press Num Pad 0 you will get a Camera View taken by the original Camera. To get Camera View from **Camera.001,** have Carcera.001 selected then press **Ctrl +**

Num Pad 0. Similarly if you have been using Camera.001 for Camera View and you select the original Camera, press **Ctrl + Num Pad 0** you get a Camera View from the original Camera.

Note: To perform Camera selection; Select Cameras in the 3D Viewport Editor or in the Outliner Editor and press Ctrl + Num Pad 0.

Select a Camera press Ctrl + Num Pad 0 For Camera View.

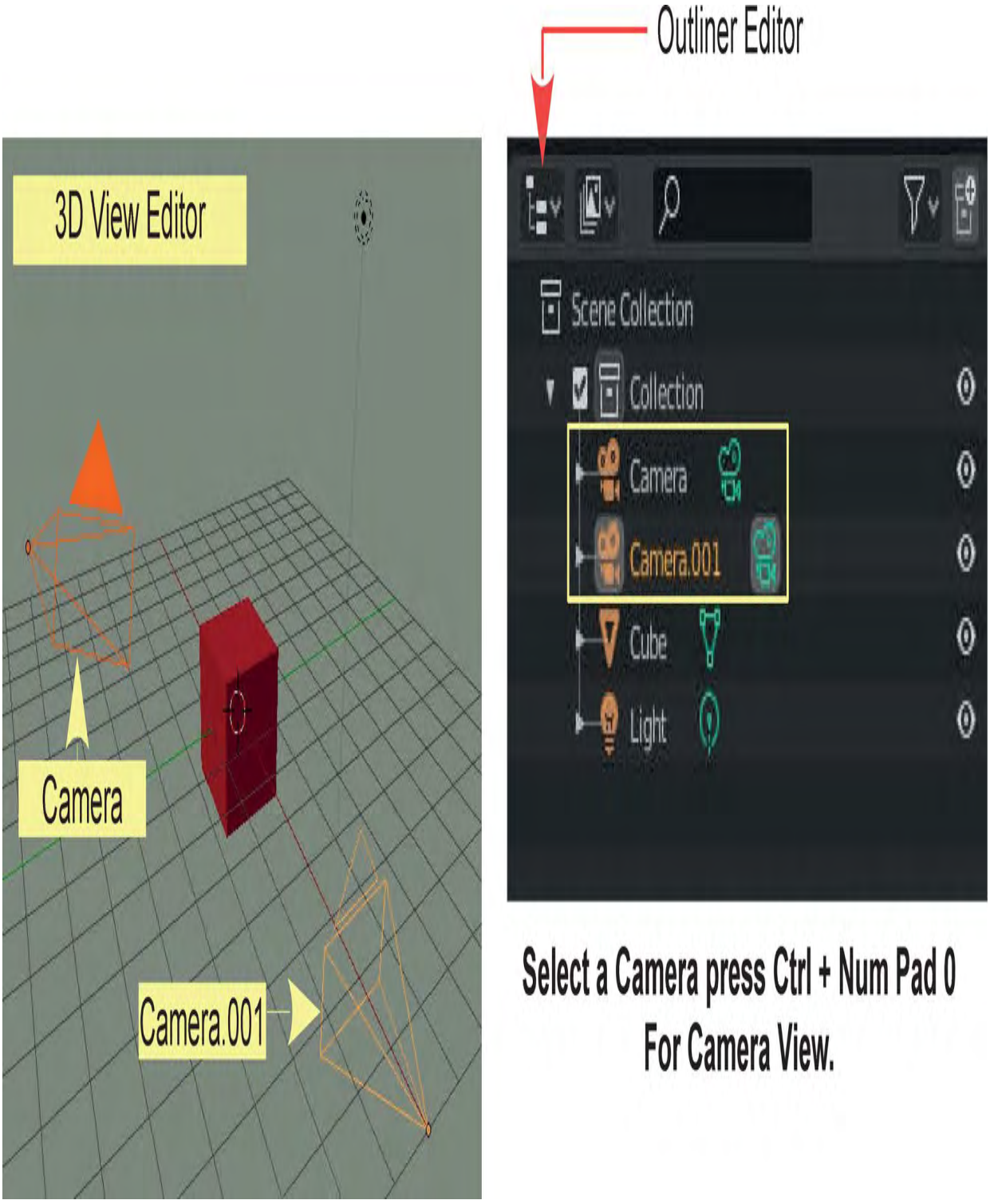
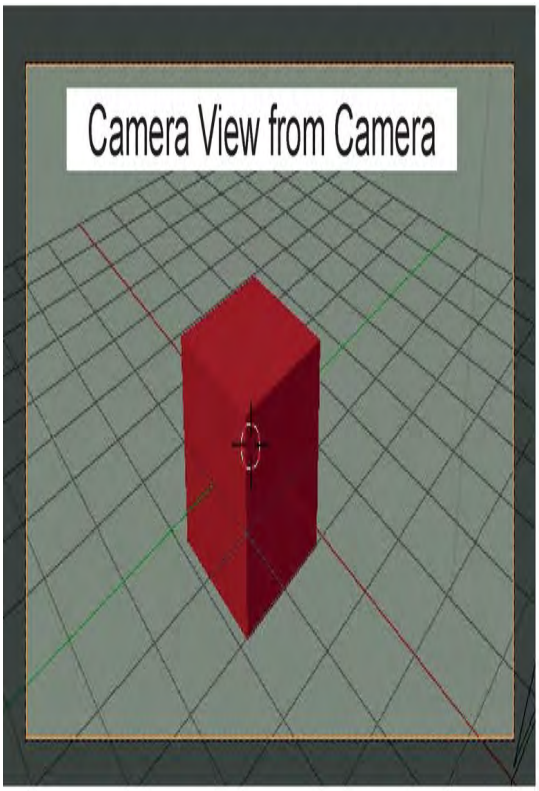
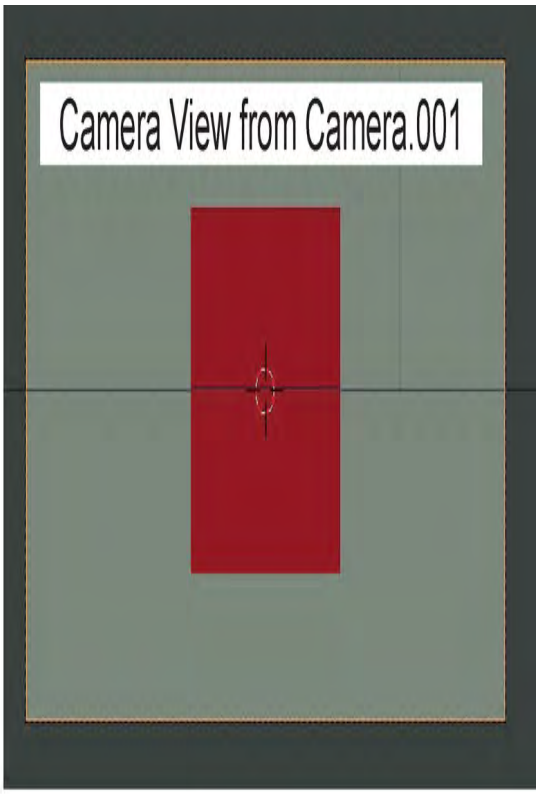

Outliner Editor
3D View Editor
Scene Collection
Collection
Camera
Camera.001
Cube
Light
Camera
Camera.001
Select a Camera press Ctrl + Num Pad 0
For Camera View.

Figure 15.8


Camera View from Camera


Camera View from Camera.001

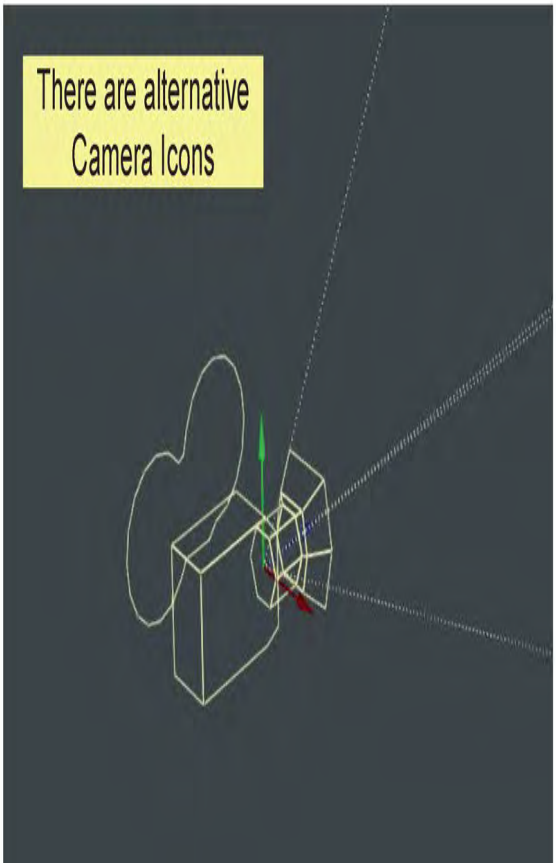

There are alternative
Camera Icons

Figure 15.9

15.4 Camera Settings

Camera settings are found in the **Properties Editor, Objec Data Properties** for the selected Camera (Figure 15.10).

By default the Lens Tab is displayed with six remaining Tabs closed.

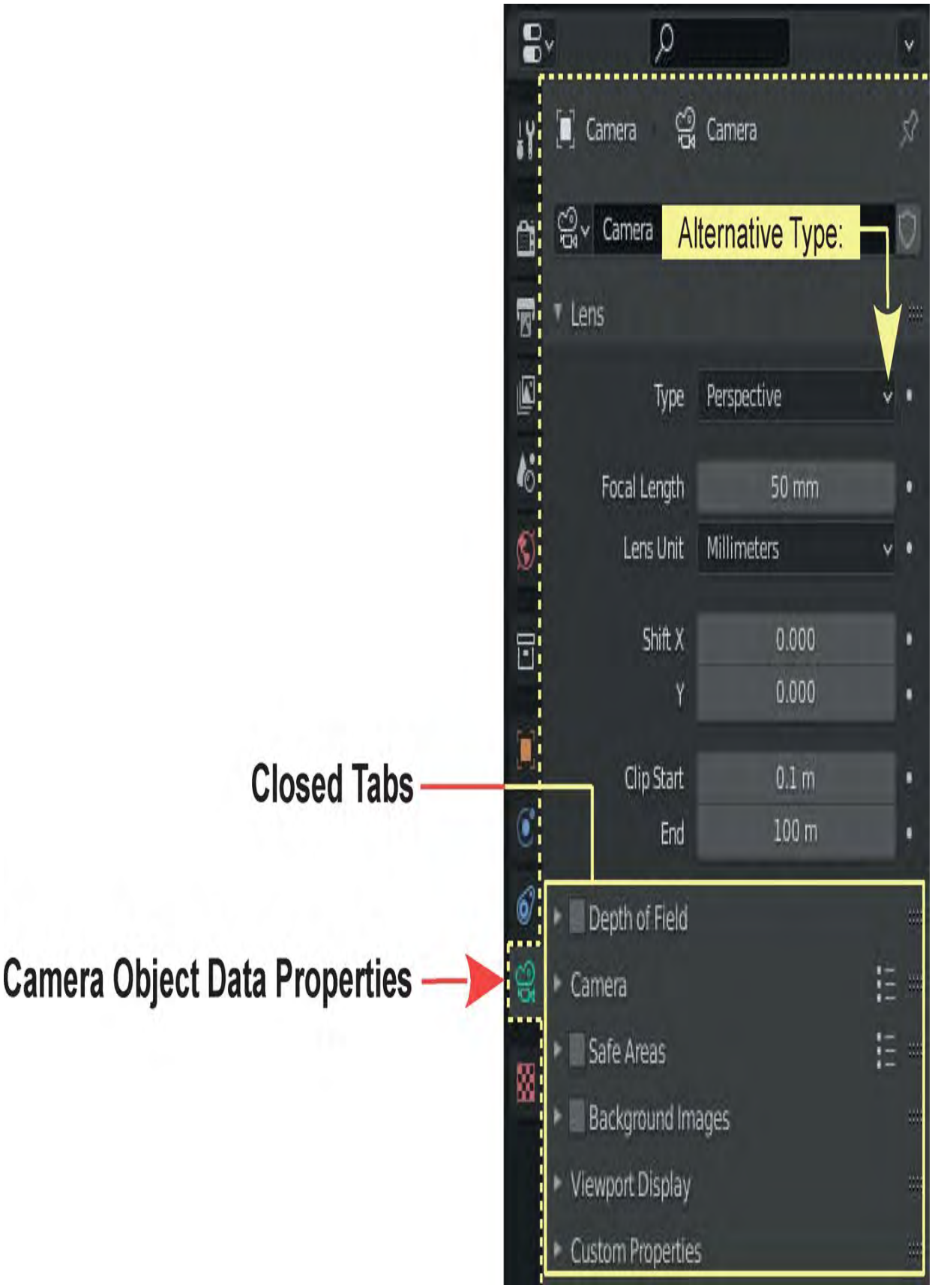

Camera
Camera
Camera
Alternative Type:
Lens
Type
Perspective
Focal Length
50 mm
Lens Unit
Millimeters
Shift X
0.000
Y
0.000
Clip Start
0.1 m
End
100 m
Closed Tabs
Camera Object Data Properties
Depth of Field
Camera
Safe Areas
Background Images
Viewport Display
Custom Properties

Figure 15.10

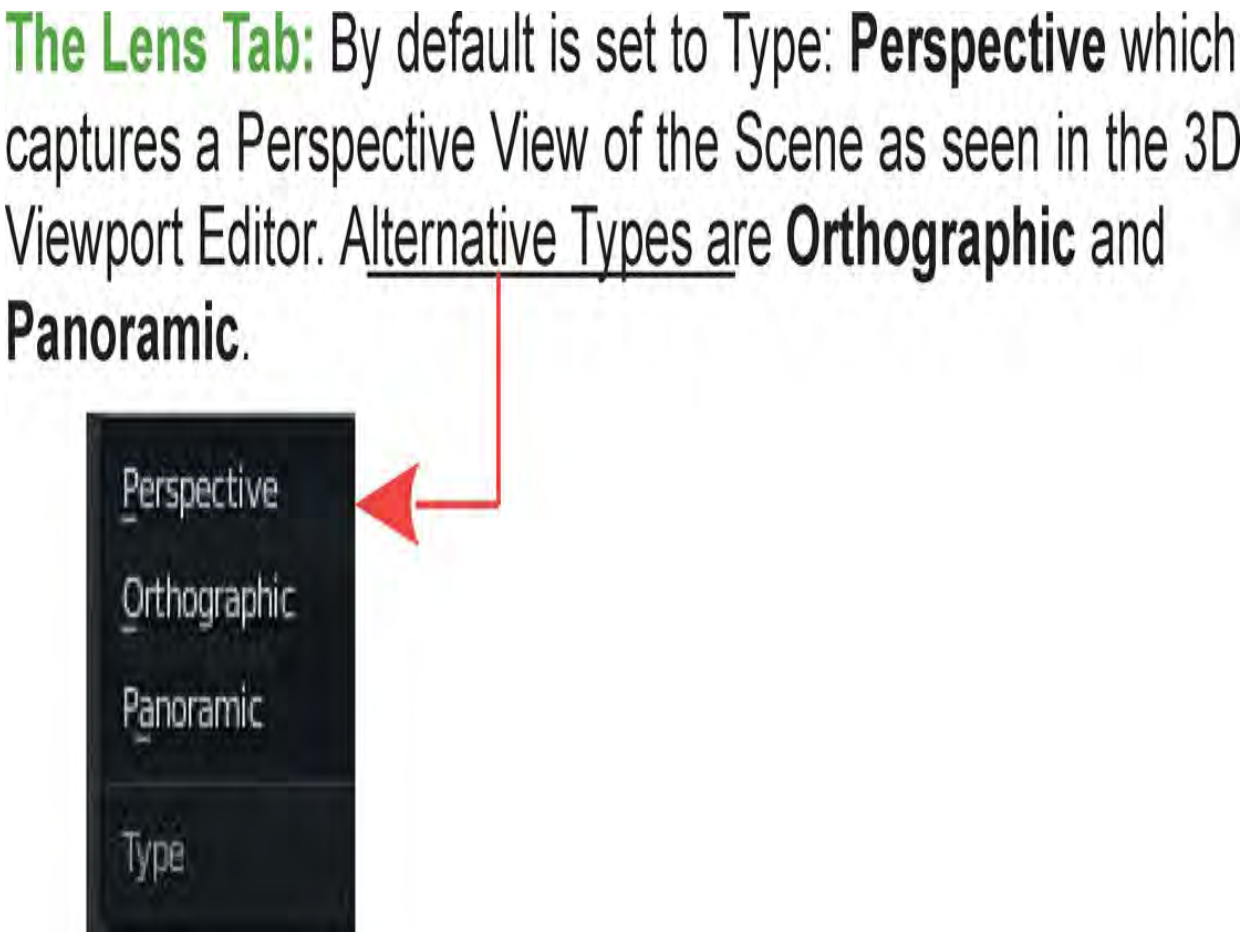


Figure 15.11

Focal Length: Sets up a lens length much like a real camera; 35mm is a good safe setting but wide and tight angle settings work for different needs.

Shift: Pushes the camera's view in a direction, without changing perspective.

Figure 15.12

Clip Start / Clip End: How close an object can be to the camera and still be seen (Figure15.13). In very large Scenes, Clip End needs to be set higher or objects disappear from view.

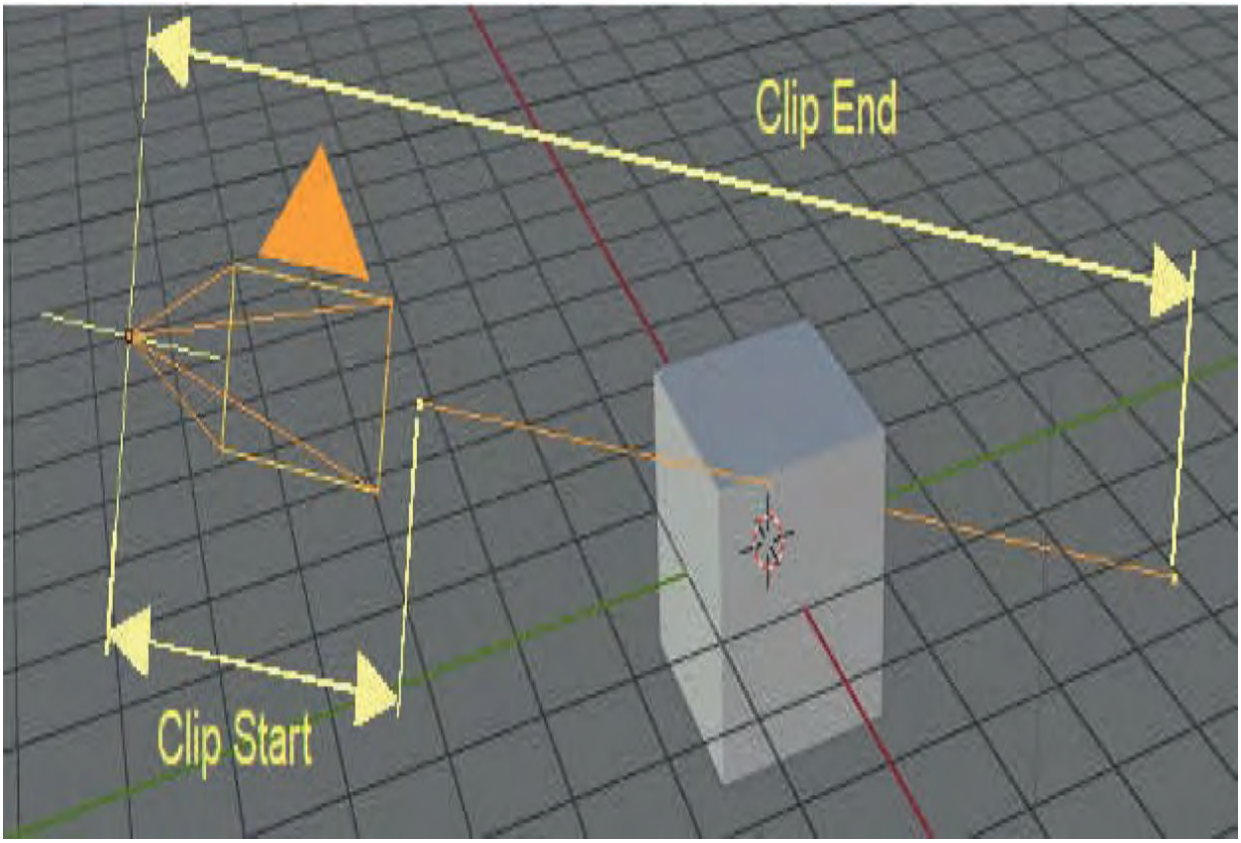


Figure 15.13

To view Limits, check **Limits** in the Viewport Display Tab (Figure 15.18)

Depth of Field Tab: (Figure 15.14)

Used with **Nodes** to blur foreground and background objects (Nodes are discussed in Chapter 16).

Aperture Tab: (Figure 15.14)

Mimics f-stop settings on a real camera which control the amount of light entering the camera.

Camera Tab: (Figure 15.15)

Camera Presets: Allows matching the virtual Camera in the Blender Scene with a real camera used to record video. This produces a more realistic effect when Camera Tracking (see Camera Tracking 15.6).

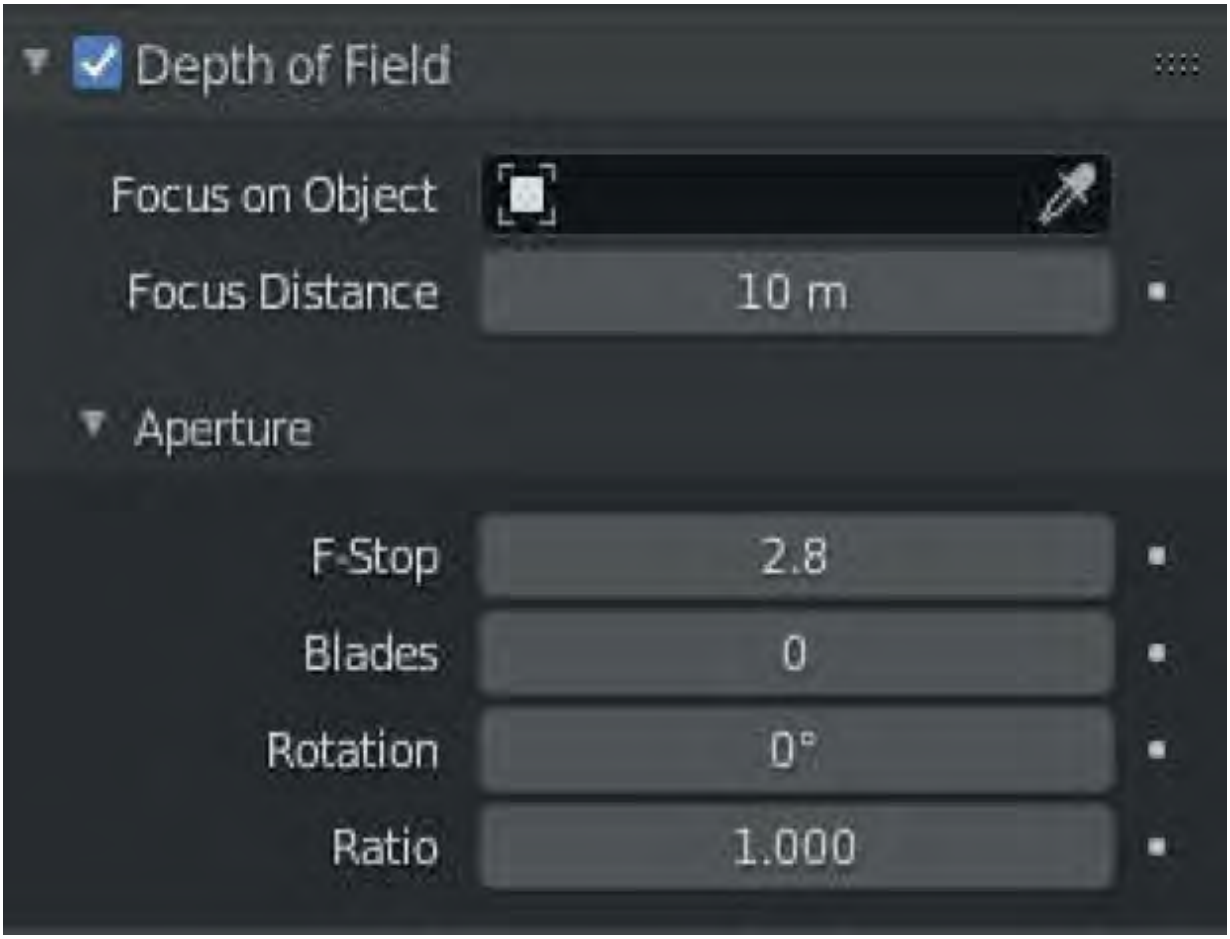
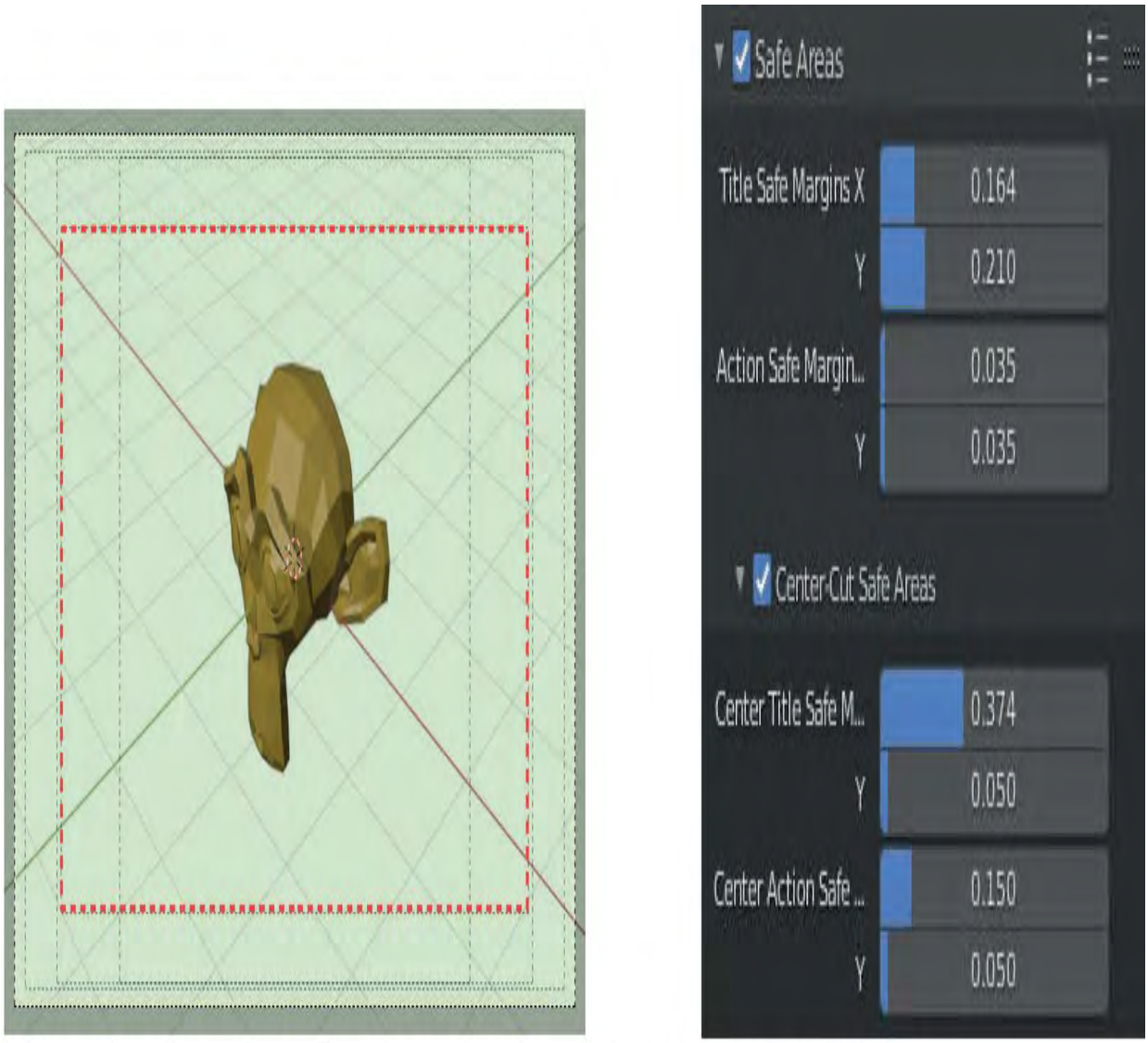


Figure 15.14

Figure 15.15

Safe Area Tab: (Figure 15.16)



Settings define areas in Camera View for inserting Captions and Titles.

Figure 15.16

Background Image Tab (Figure 15.17)

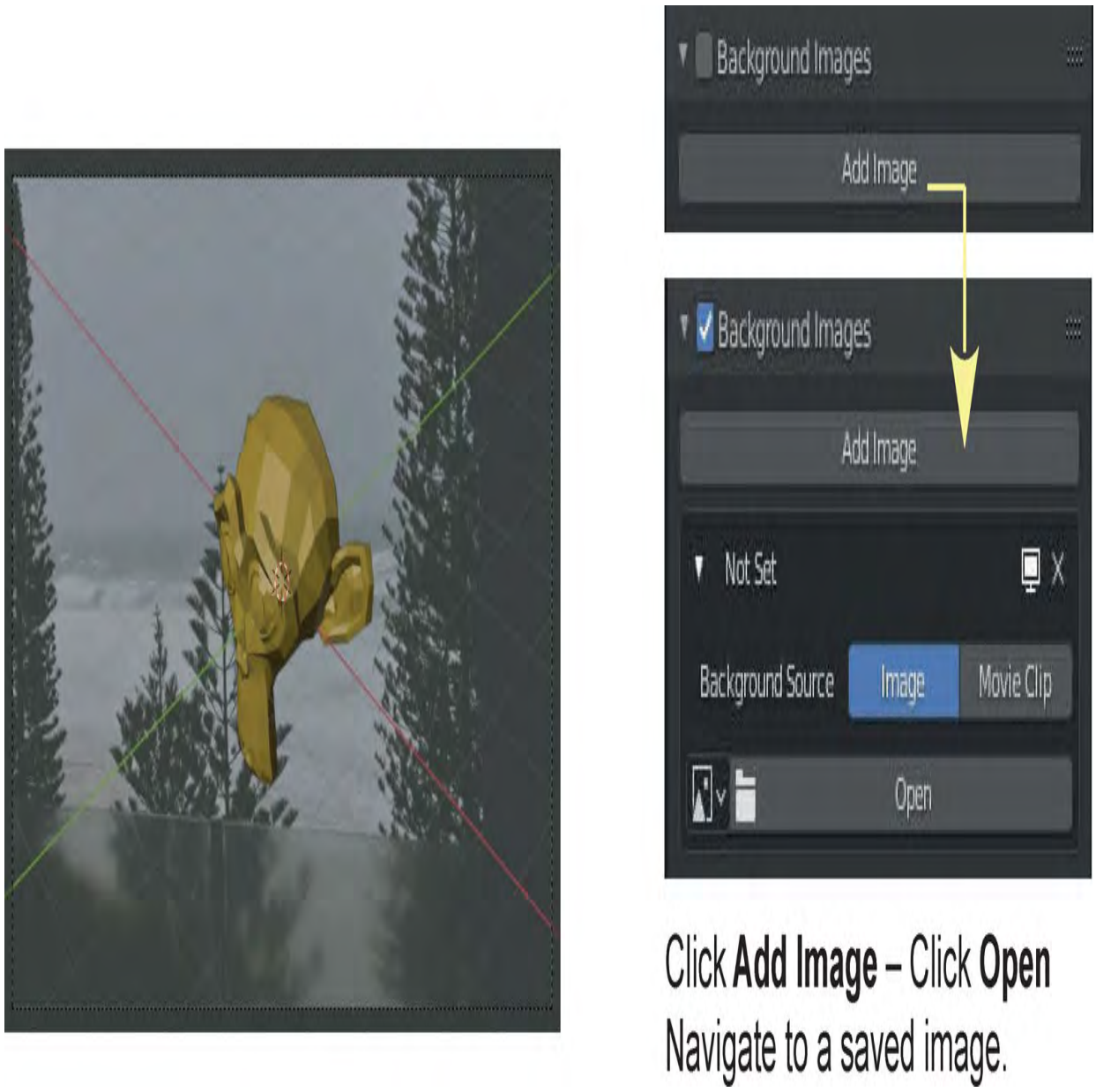


Figure 15.17

Inserts a reference image or movie clip in the Camera View in the 3D Viewport Editor. **The image does not render**

Viewport Display Tab (Figure 15.18)

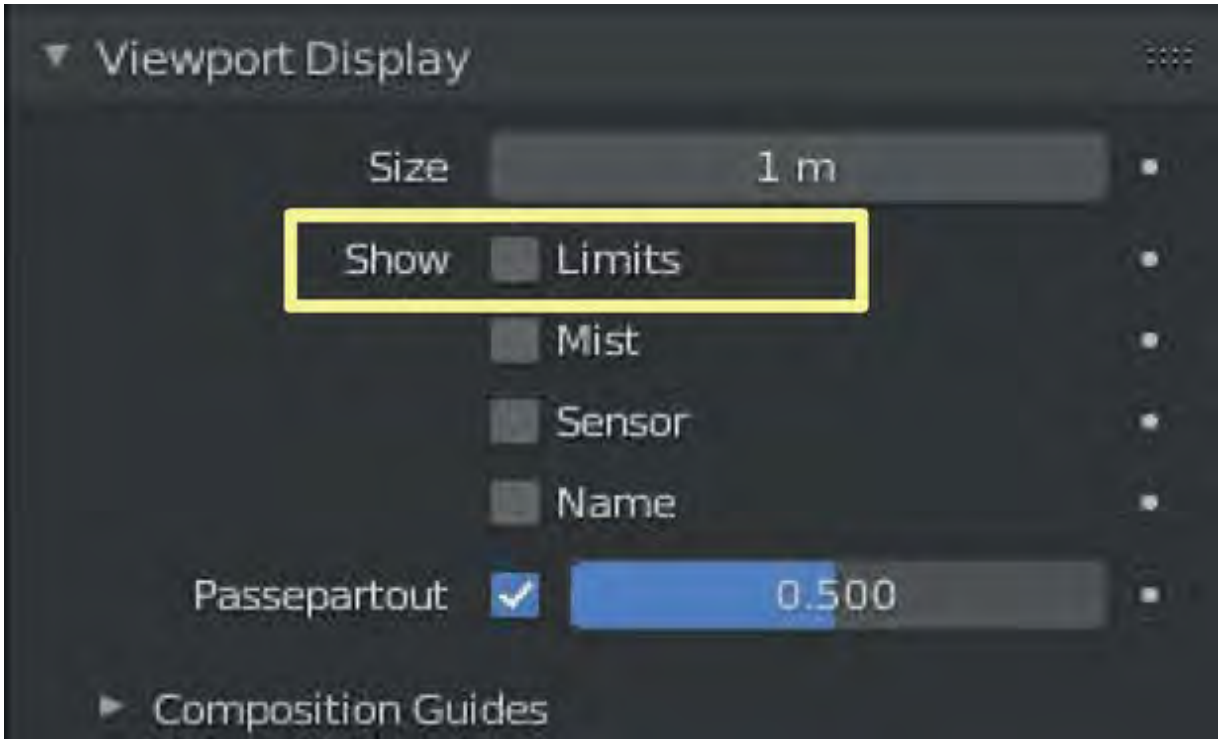


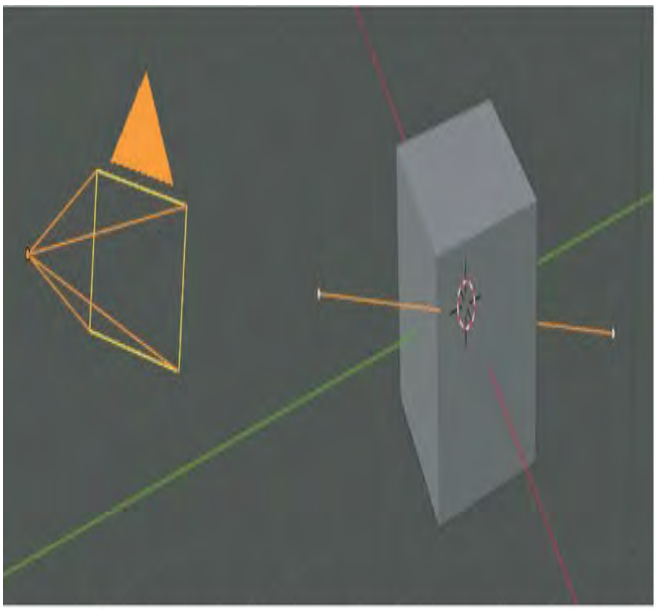
Figure 15.18

Viewport Display options determine what you see in the 3D Viewport Editor.

Size: How big to draw the Camera on the Screen; you can also control the size with scale.

Limits: Draws a line in the scene to help you visualise the Camera's range (Figure 15.19). The Limit is set by the **Clip Start** and **Clip End** values in the **Lens Tab.**

Mist: Gives you a visual display of how far the Camera sees if you are adding a mist effect.



Not to be confused with the Limits, Clip Start and End for the Camera.

To create a Mist Effect See: Volumetrics 15.10

Figure 15.19

Sensor: Shows the sensor size (Film Gate) in Camera View (Figure 15.20).

Name: Displays the name of the active Camera in Camera View (Num Pad 0) (Figure 15.20).

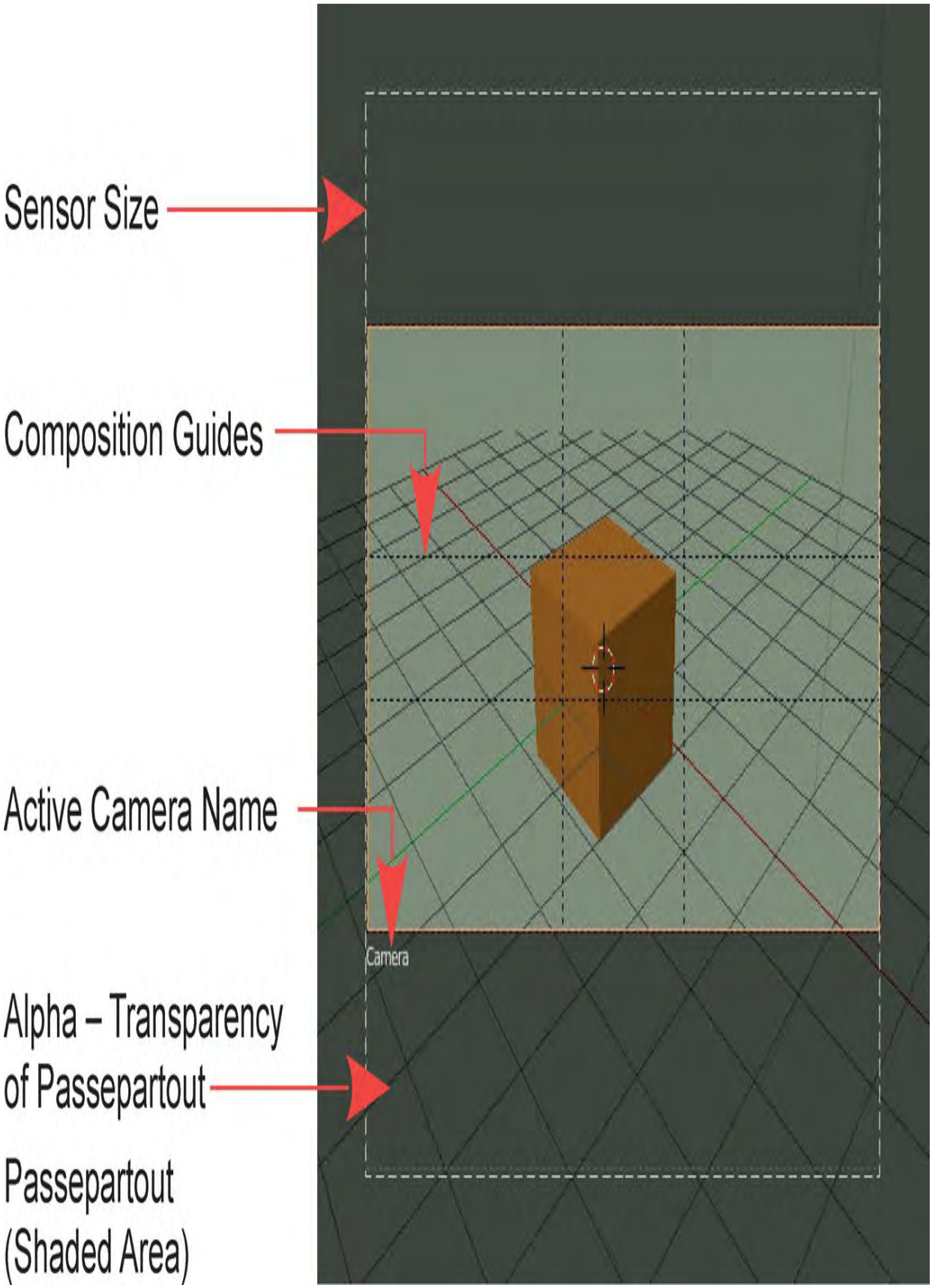

Sensor Size
Composition Guides
Active Camera Name
Alpha – Transparency of Passepartout
Passepartout (Shaded Area)
Camera

Figure 15.20

15.5 Camera Switching

In **Cameras 15.3** it was shown that you may have more than one Camera in a Scene and you can switch between Cameras by selecting one, then pressing **Ctrl + Num pad 0.** This makes the selected Camera active and opens Camera View showing what is seen by that Camera. Manual selection is fine for rendering single images of an object from different viewing perspectives but you may want to Animate the switching such that when the animation plays you switch between Cameras.

Animation is discussed in Chapter 20 but a prelude in the following demonstration will be beneficial.

Set up a Scene similar to that shown in Figure 15.21, with three **Cameras** pointed at **Suzanne** (Monkey) from different locations. You can use the default Camera and add two others. You will have to Rotate and point each Camera at Suzanne and check Camera View for each. **Remember:** With more than one Camera in a Scene you press **Ctrl + Num Pad 0** for Camera View with a camera selected.

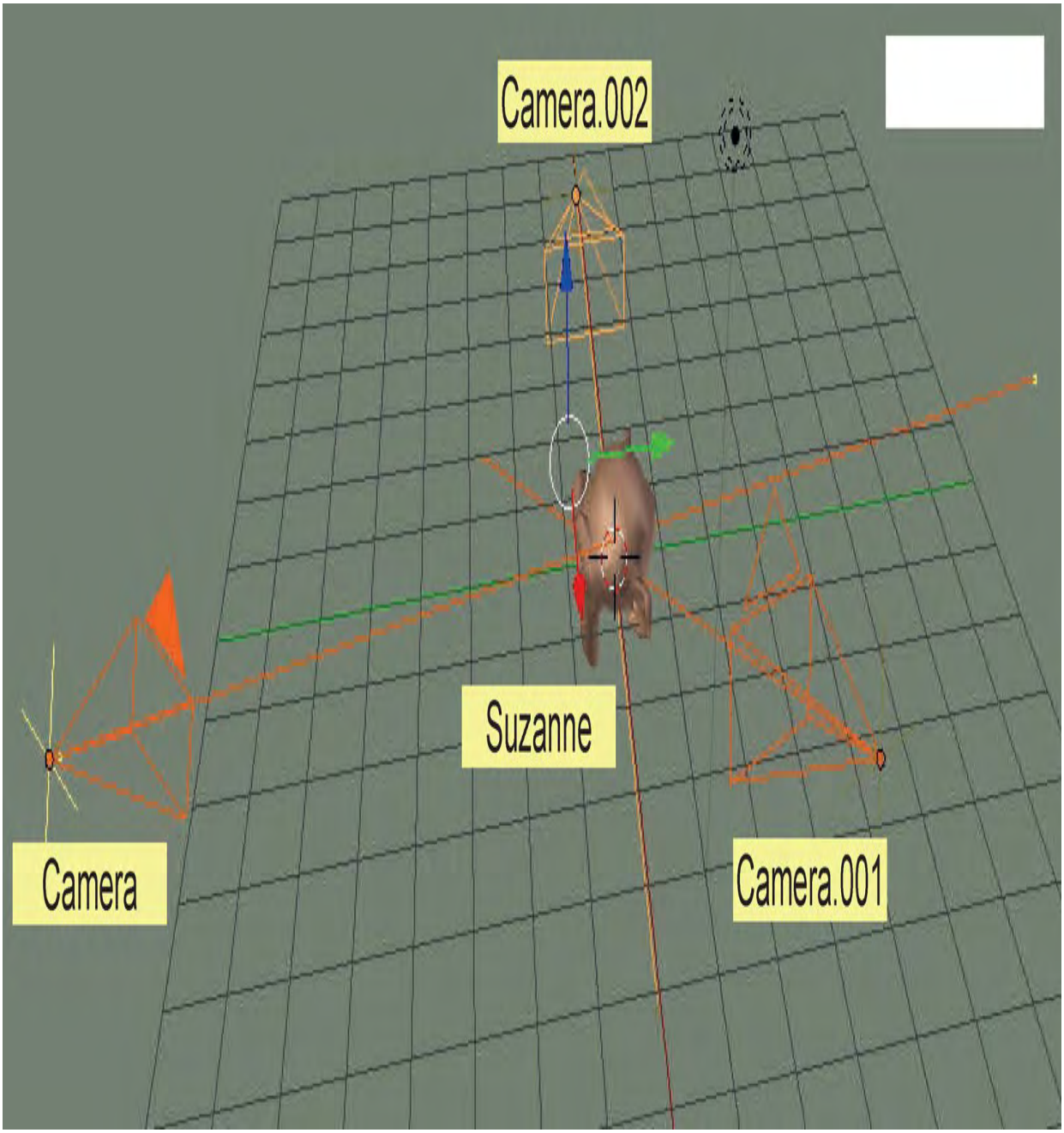


Figure 15.21

In the 3D Viewport Editor select the default Camera named **Camera. In** the **Timeline Editor** with the vertical blue cursor line at Frame 1 and the **Mouse cursor in the Timeline Editor** press the **M Key** to place a **Marker** at Frame 1. **Note:** You can only place the marker when the **Mouse Cursor is positioned** in the Timeline (Figure 15.22).

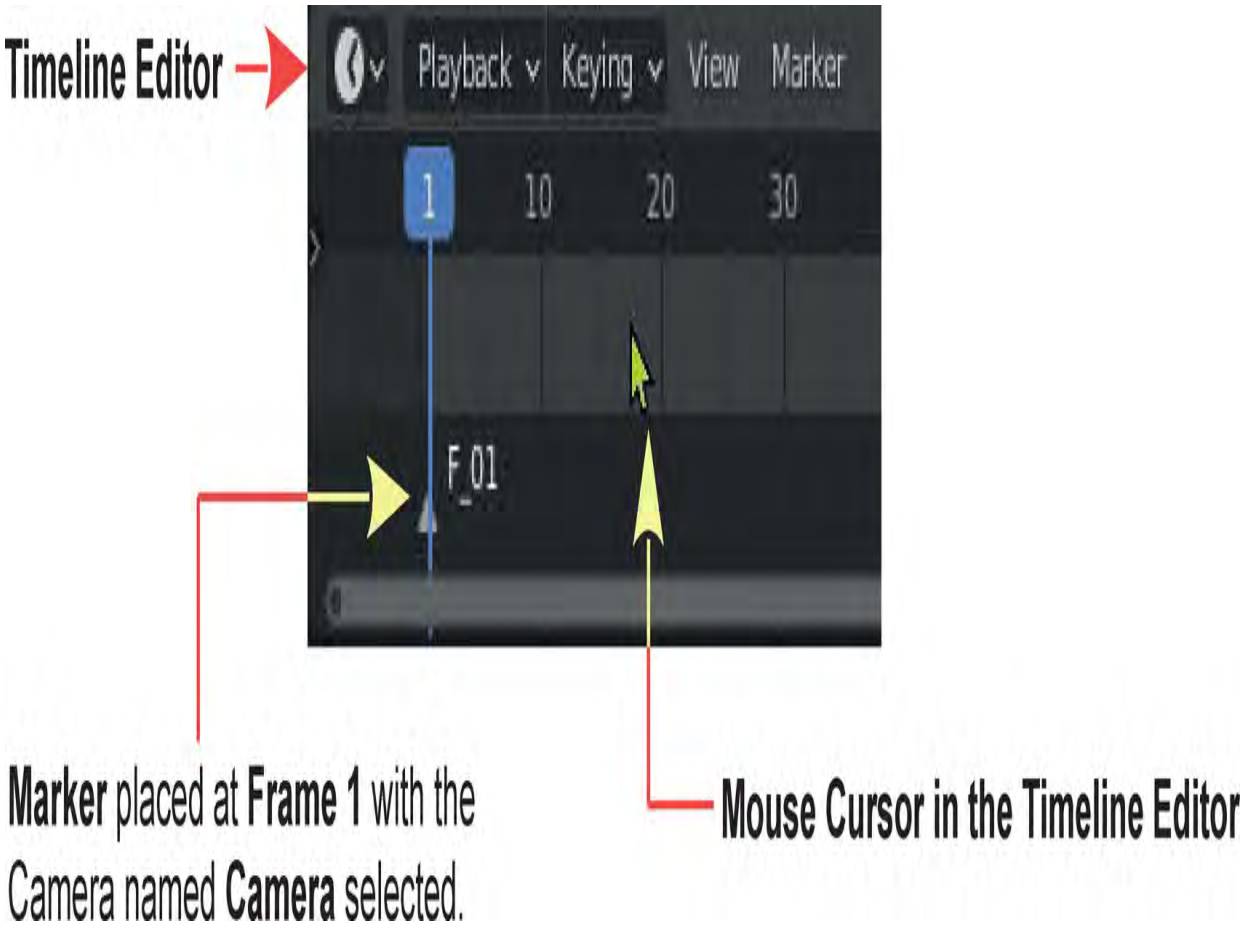


Figure 15.22

Deselect **Camera** and select **Camera.001.**

With Camera.001 selected, move the Timeline Cursor (blue line) to Frame 40. Press the M Key to insert a Marker. Repeat for Camera.002 at Frame 70. Note: It is not necessary to have Camera View for each Camera displayed when placing Markers.

At this point nothing happens in the 3D Viewport Editor when the Animation is played in the Timeline. **Each Camera has to be Bound to a Marker.**

Select **Camera** in the 3D Viewport Editor, click on **Marker F_01** in the Timeline Editor (highlights white), click **Marker** in the Timeline Header and

select **Bind Camera to Marker** (alternatively Press Ctrl + B Key). The Camera is Bound and the Marker in the Timeline Editor shows as

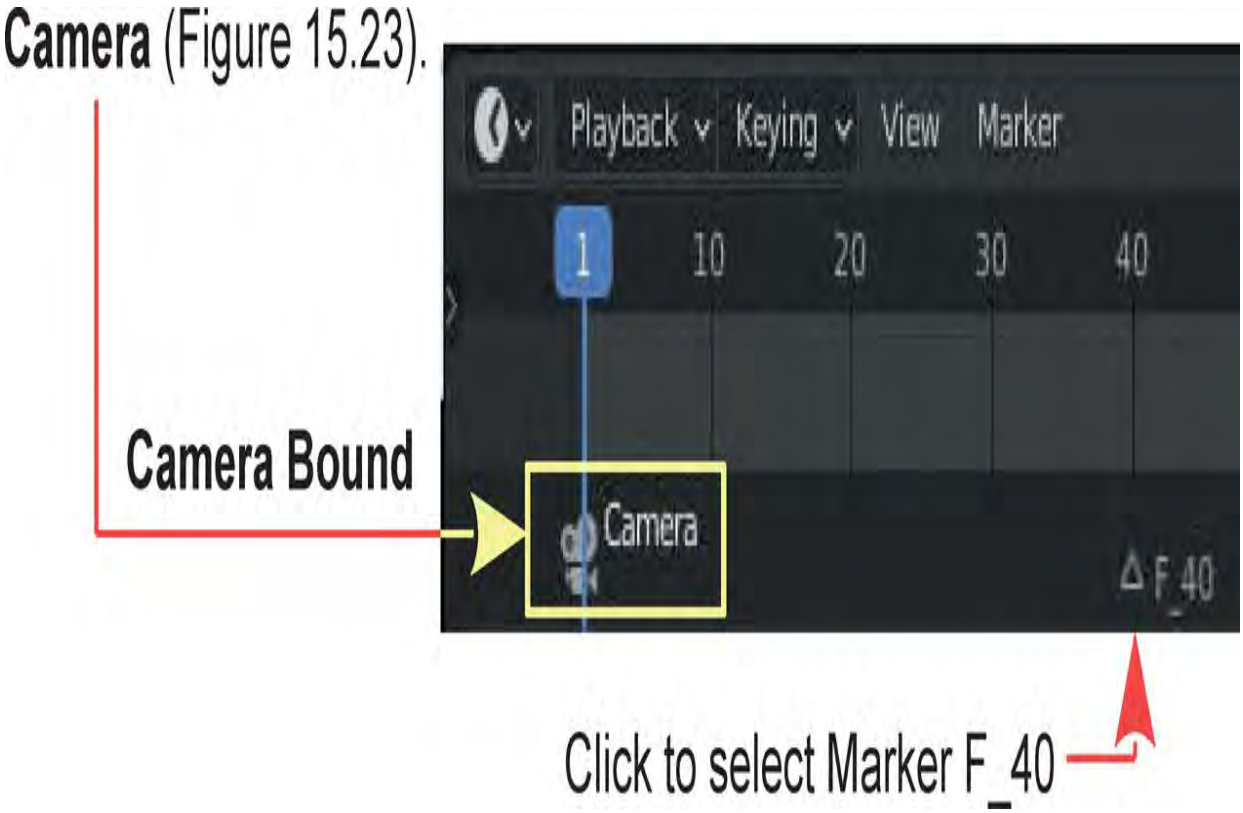
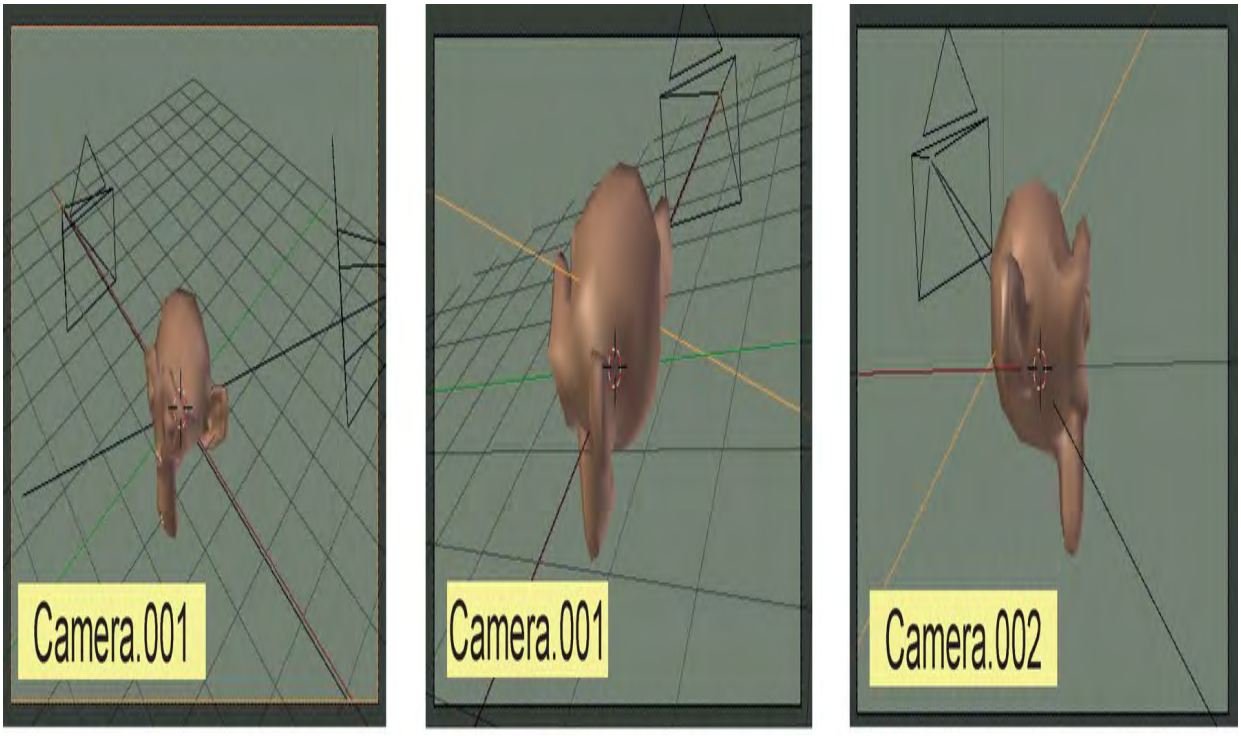


Figure 15.23

Deselect Camera, select Camera.001, move the Timeline Cursor to Frame 40 (the position of Marker F_40), Press Ctrl+B Key to Bind Camera.001 to the Marker. Repeat for Camera.002.

Figure 15.24

When the Animation is played in the Timeline Editor, Camera View changes in the 3D Viewport Editor as the Timeline Cursor passes the Frame where a Marker is set.



Camera View F1 to F40 Camera View F40 to F70 Camera View F80 to F250

Figure 15.25

The Animation plays from Frame 1 to Frame 250 then repeats.

15.6 Camera Tracking

Camera Tracking is a technique that imitates the real Camera motion which occurs when recording a video. This motion is applied to a 3D Camera in a Blender Scene providing a realistic effect when a 3D model is superimposed over a video background (Figure 15.26). Without this effect the Blender 3D Camera would track to a stable imaginary point or to a predetermined curve track in the Scene. This would be fine for the superimposed 3D Object but the actual video used as a background would move differently and produce an unrealistic effect. The essence of the technique is to plot the movement of multiple points in the video Scene and feed that information to the motion of the Blender 3D Camera.

At this point the technique is mentioned to make you aware of its existence and the following video tutorial is suggested: https:www.youtube.com/watch?v=O3fGc QM3yl//w

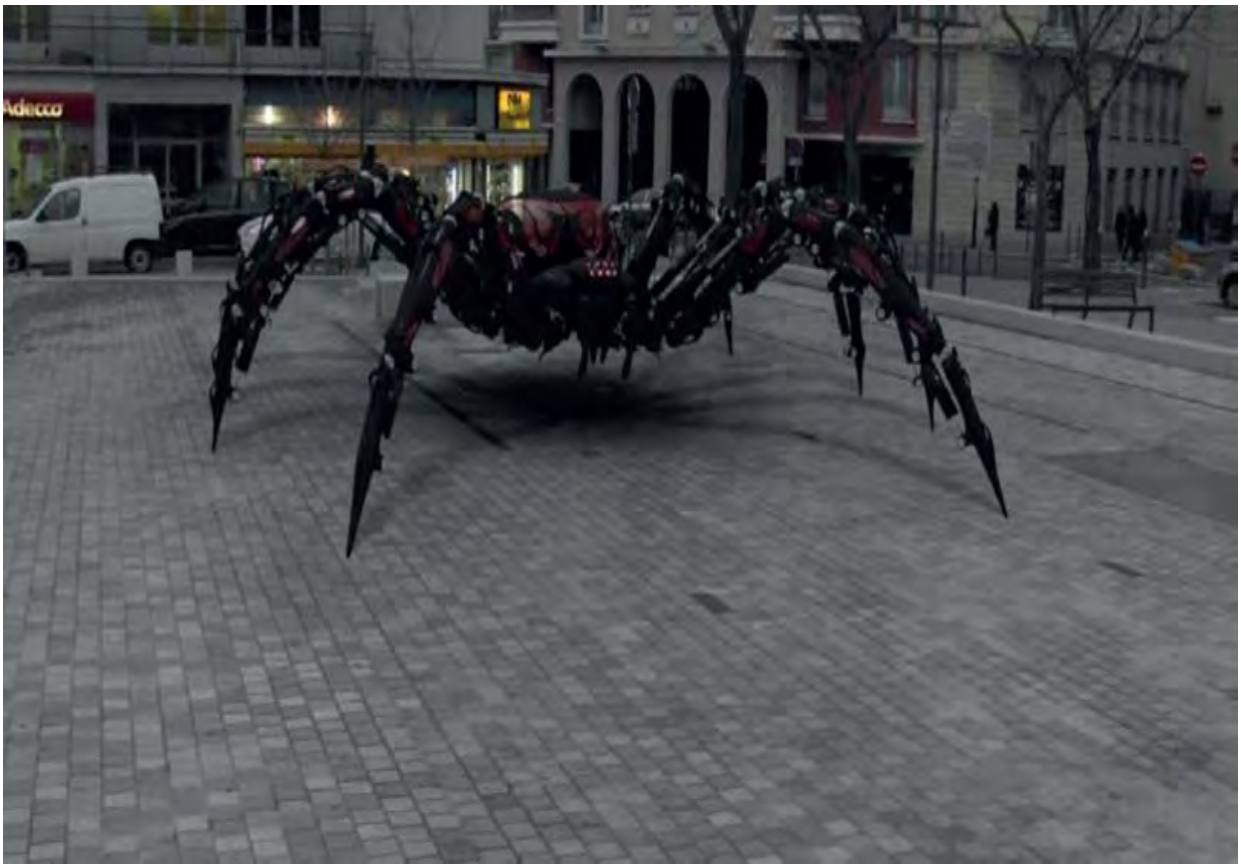


Figure 15.26

15.7 Basic Lighting Arrangement

Scene Lighting (15.1) at the beginning of the chapter demonstrated how the illumination from Lights in the Scene influences what is captured by the Camera and ultimately Rendered.

To illuminate a Scene will usually require multiple Lights to create a mood by highlighting content and controlling shadows. To demonstrate very basic Scene Lighting the Scene arrangement created in 15.2 – Figure 15.5 will be used with the application of Material color to the Objects (Chapter 1 – 1.12 Blender Node System disabled).

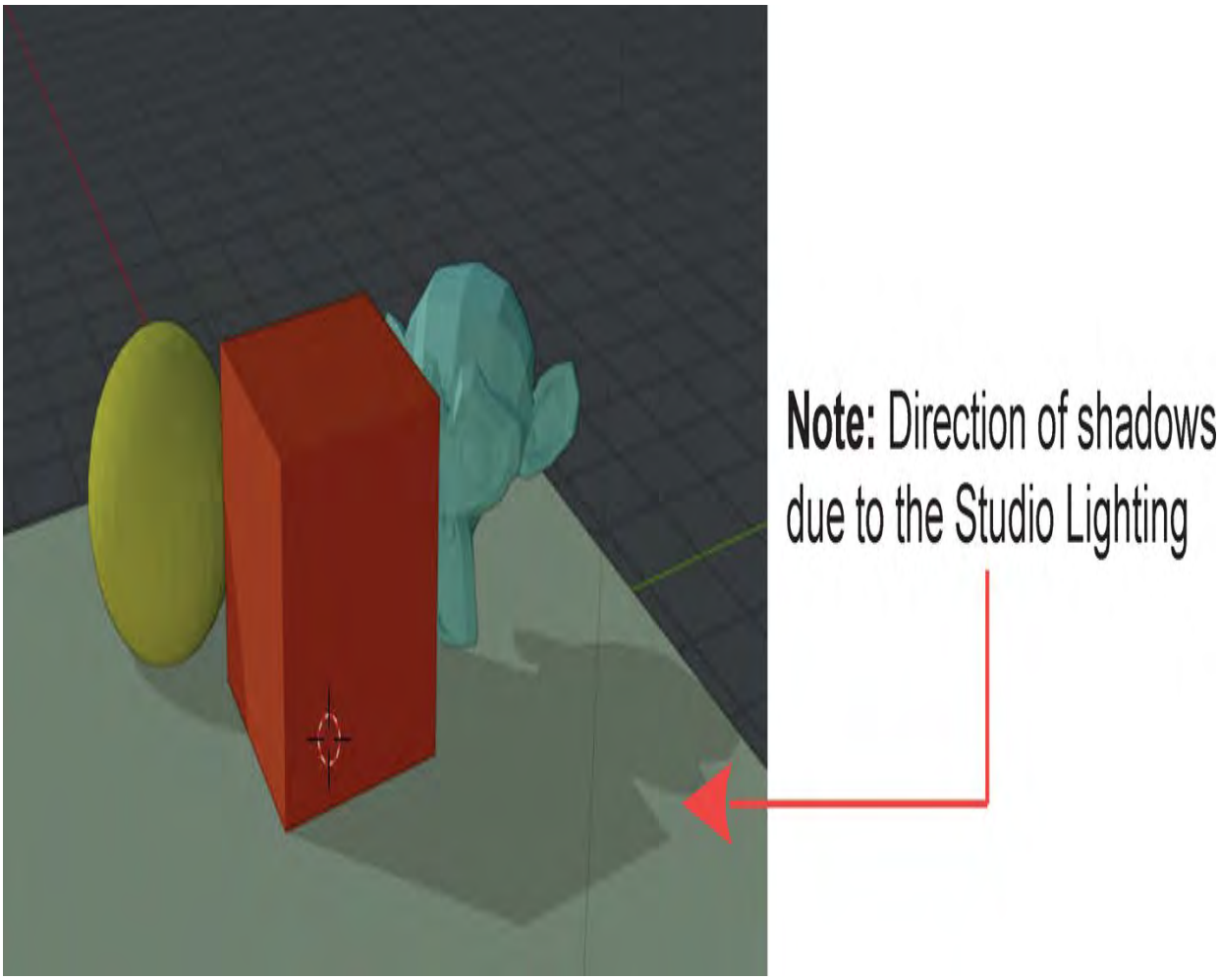


Figure 15.27

Solid Viewport Shading

You may create the Scene with the 3D Viewport Editor in **Solid Viewport Shading Mode** (Chapter 14). The default settings in the Sub Options Panel may be left as they are, with the one exception, **check Shadows. Remember:** The Point Light has no effect when in Solid Viewport Shading Mode.

Note: In Solid Viewport Shading Mode, Sub Options, the default color setting is: **Color - Material,** meaning the Material (color) applied to the Objects will be displayed.

Add a basic Material (color) to all Objects (Chapter 4 - 4.18).

Figures 15.27 – 15.28 are screen captures of part of the view in the 3D Viewport Editor approximating Camera View (what the Camera sees).

With the Scene created, change to **Material Preview, Viewport Shading Mode** (Chapter 14) and check **Scene Lights** and **Scene World** in the Sub Options panel. You will immediately see a difference in the view since the effect of the single default Point Lamp is displayed.

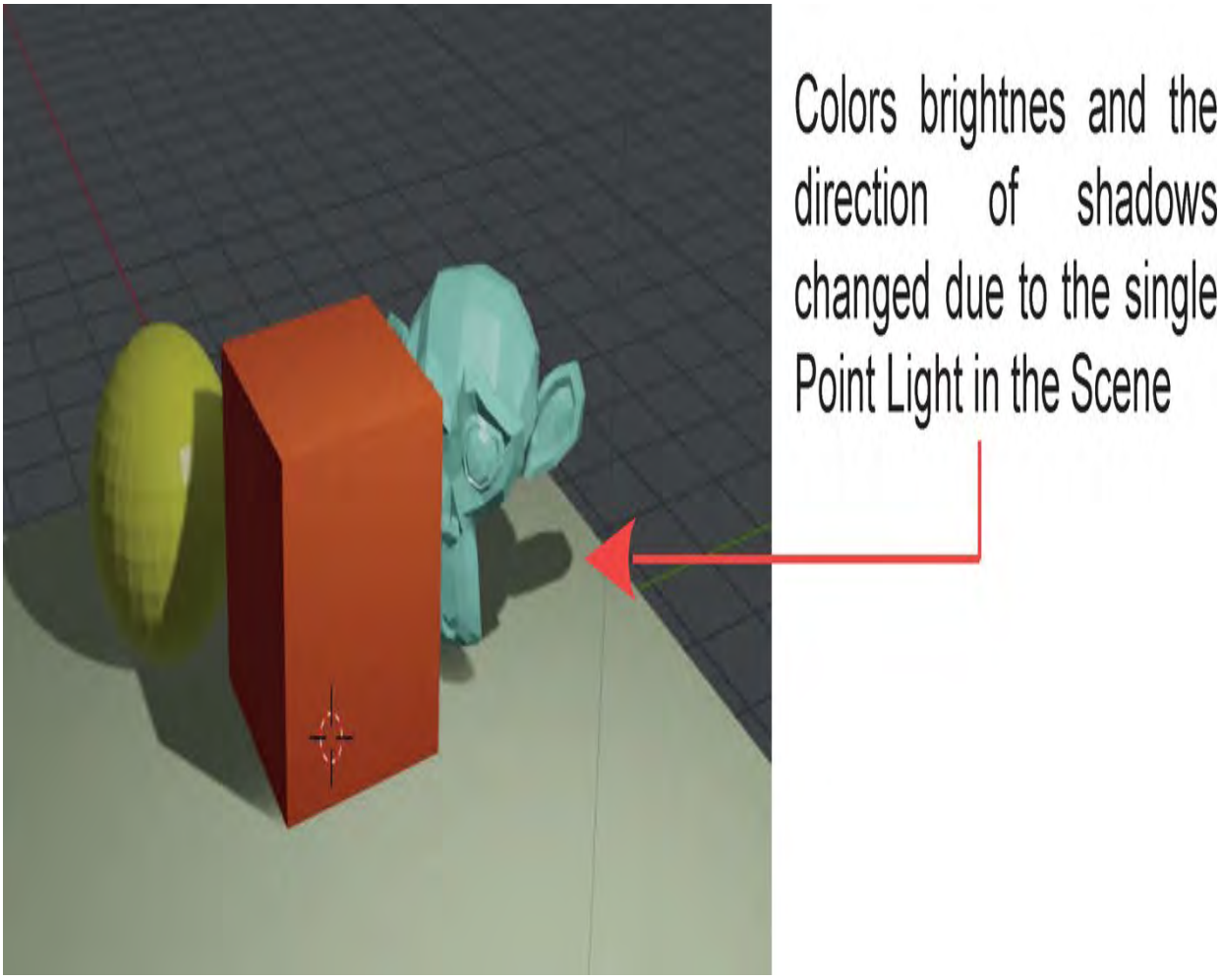


Figure 15.28

Make note that the Light emitted by the Point Light, by default, is white (RGB 1.000).

In the **Outliner Editor** click on **Light** to select the Point Light in Collection. Selecting in the Outliner Editor selects the Light in the 3D Viewport Editor.

In the **Properties Editor, Object Data Properties** the settings will be applicable to the Point Light in the 3D Viewport Editor (the selected Object). Change the Light setting Point to **Area.** Again you see a change to the lighting in the 3D Viewport Editor demonstrating the effect of Light settings (Figure 15.29).

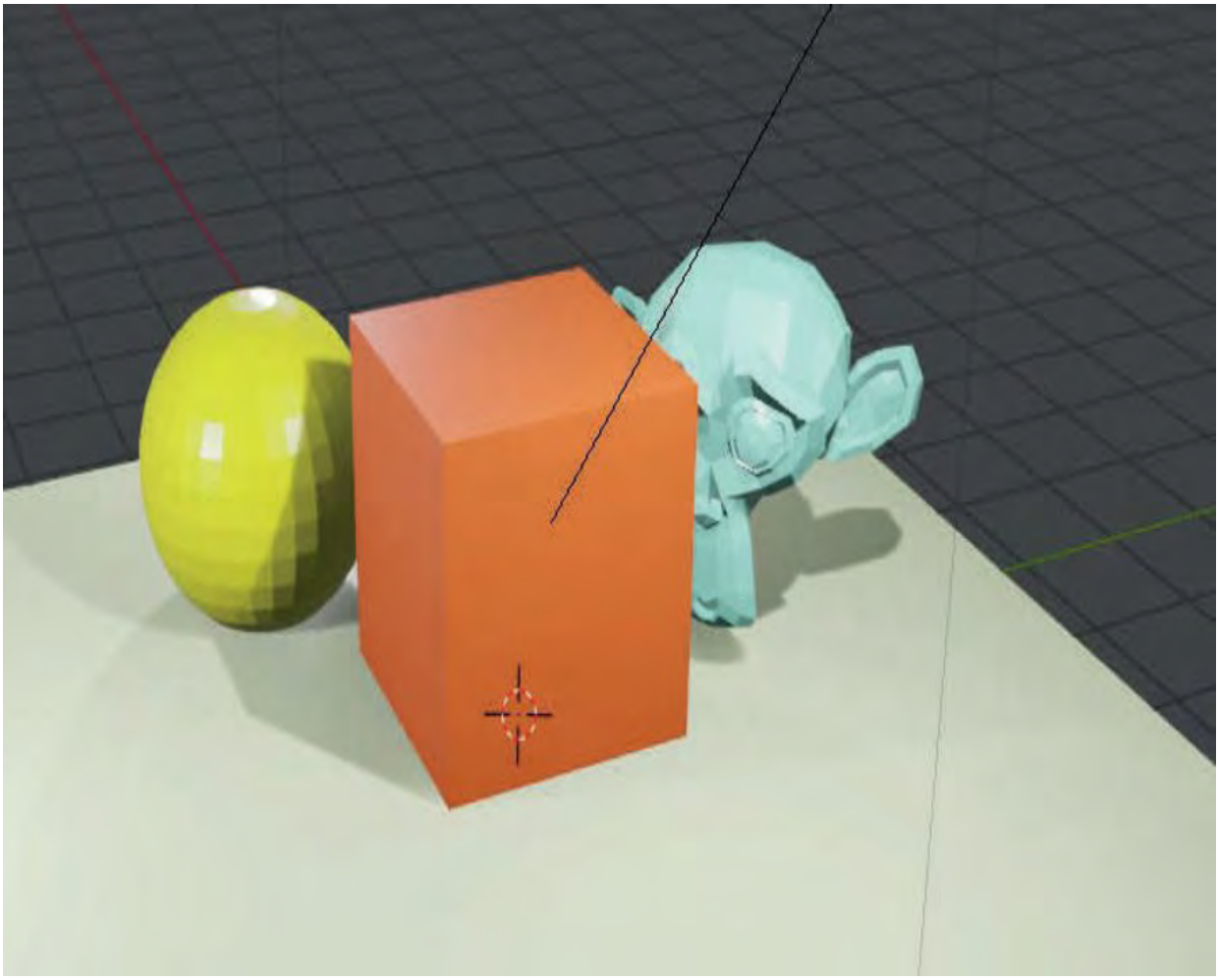


Figure 15.29

Note: What you see on your computer screen will probably differ to the Figures being shown due to differences in arrangement and settings. The objective here is to demonstrate technique.

In the 3D Viewport Editor add two more Point Lights and position as shown in Figure 15.30.

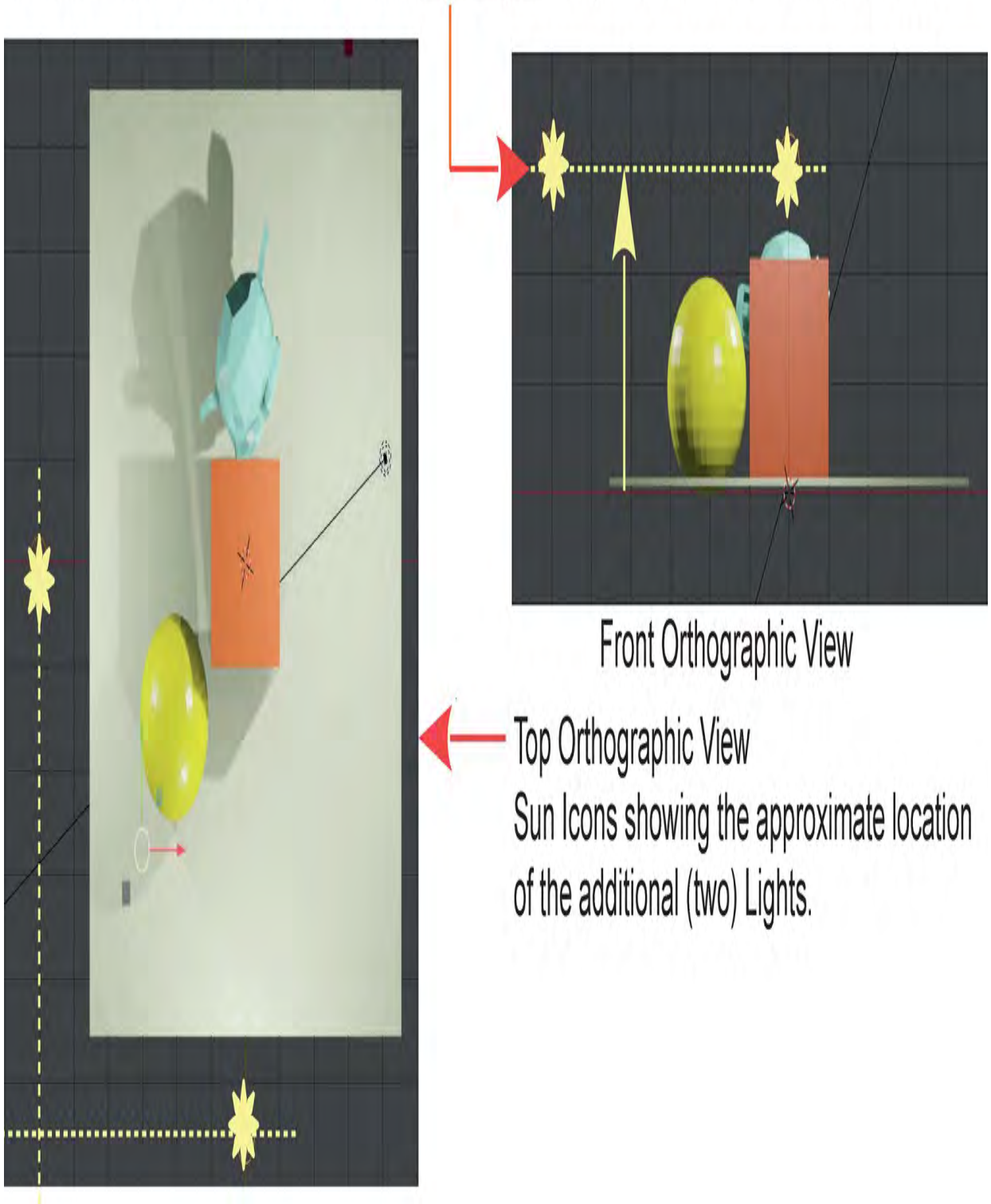
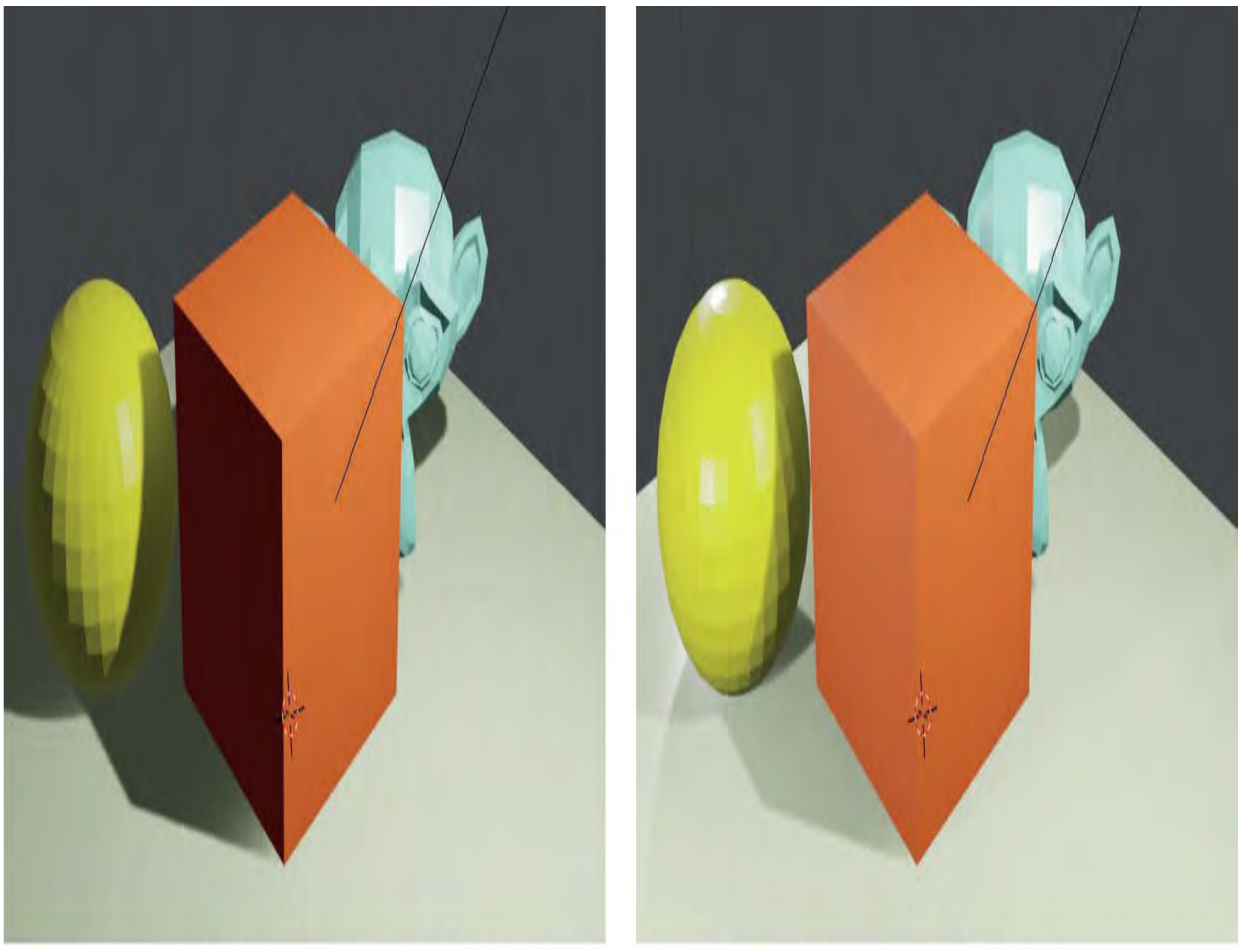


Figure 15.30

With the 3D Viewport Editor in **User Perspective View** go to the **Outliner Editor** and toggle hide and display of the new Lights to see the different Lighting effects.

Note: The new Lights listed in the Outliner Editor are named **Point** and **Point.001**



Area Light only, Point and Point.001 Hidden (off) Area Light + Two Point Lights

Figure 15.31

15.8 Background Scene Lighting

Beside the effect of Lights in the Scene you should be aware that the Scene Background plays a part in determining how Materials display. As in the real world the ambient light surrounding Objects determines how they are seen. In Blender, the World Background light has a similar effect.

Background Color Lighting

To demonstrate this phenomena start a new Blender File and replace the Cube Object in the default Scene with a UV Sphere and set the Sphere's surface to Shade Smooth (Chapter 4 – 4.21 Figure 4.35).

Have the 3D Viewport Editor in **Rendered Viewport Shading Mode** (Chapter 14). With the UV Sphere selected, In the Properties Editor, Material Properties, click **New** to add a Material. Since this is a new Object in the Scene, **Use Nodes** will be activated.

Use Nodes (highlighted blue), means that the **Blender Node System** for applying Materials is being used. An explanation describing what Nodes are and how to use them is given in Chapter 16.

For the moment, with **Nodes active,** click on the **Base Color** bar to display the Color Picker Circle and give the UV Sphere a yellow color by setting the RGB color values (R: 0.753, G: 0.531, B: 0.002). Have the intensity slider cranked all the way to the top (very bright).

Remaining in the Properties Editor, click on the **World Properties button.** In the **Surface Tab, Use Nodes** is active. By default **Surface** is set as **Background.** Click on the Color bar (the default color is gray) and change to a blue color (R: 0.011, G: 0.789, B: 0.900). Move the Intensity Slider up to about half way.

Result: Blue Background causing the Yellow Sphere to appear Green especially in the shadow.

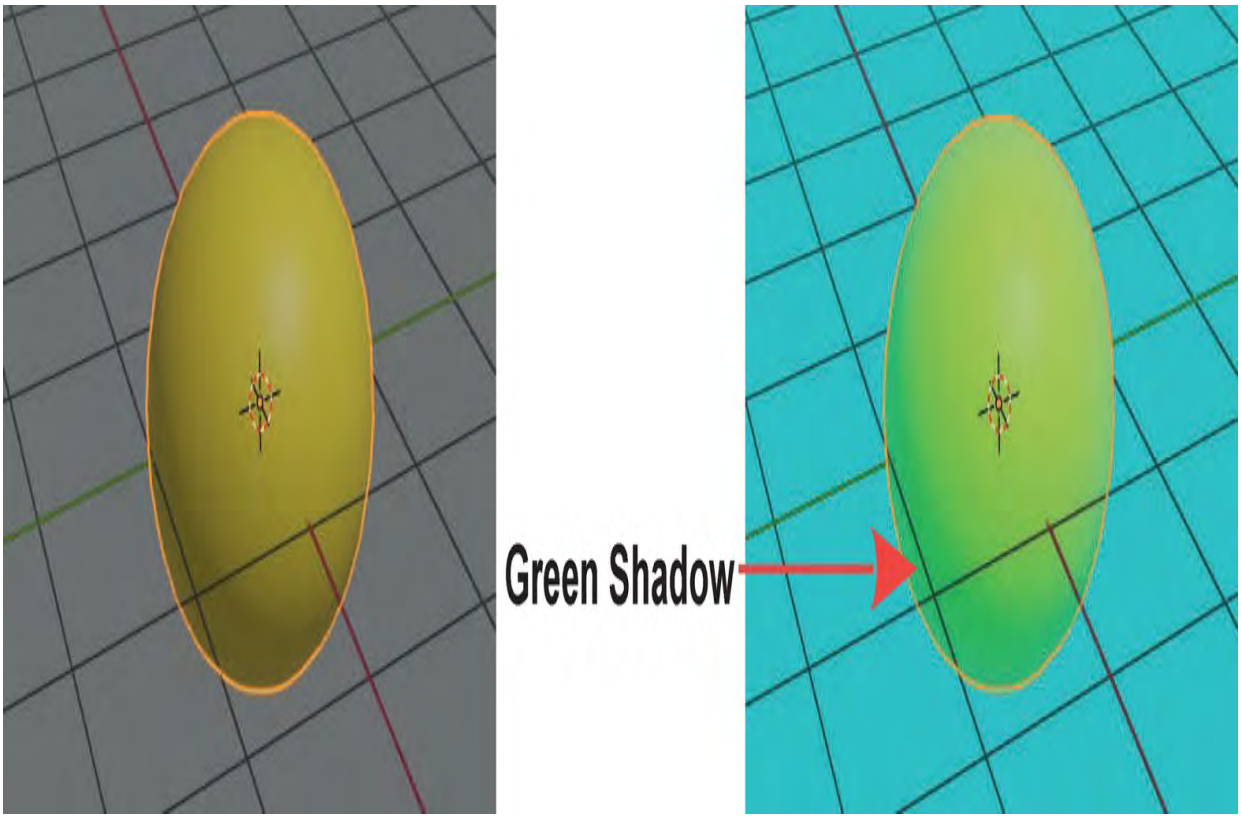


Figure 15.32

15.9 Images as a Background

HDRI Maps

Colored Scene backgrounds are suitable in many cases but an image used as a background can significantly add atmosphere. Special images called **HDRI Maps** are particularly spectacular in giving a three dimensional effect when the Scene is rotated. This type of image adds lighting to the Scene thus affecting how Objects are seen.

One source of HDRI images is: https://hdrihaven.com

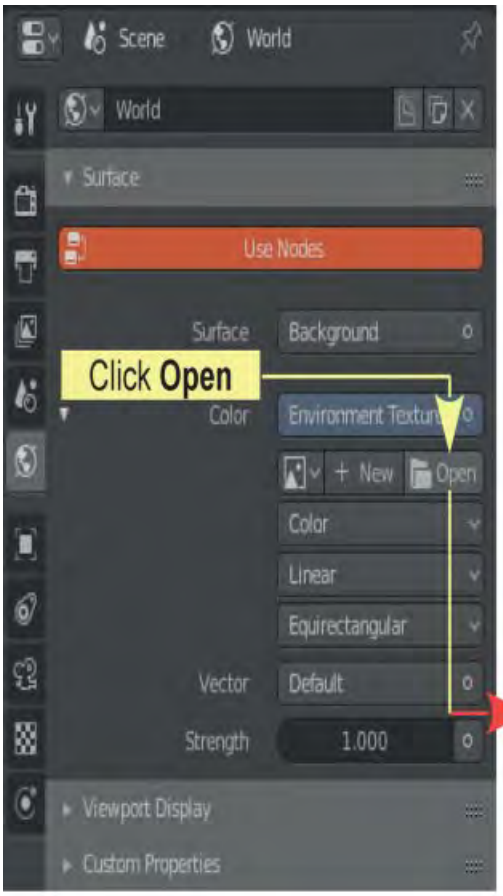
Figure 15.33

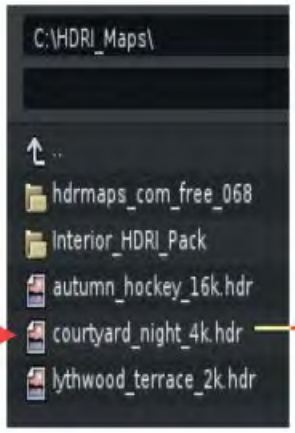
To demonstrate the use of HDRIs use the Scene arrangement in Figure 15.26. Deselect all the Objects in the Scene and have an **HDRI image** saved on your computer.

In the **Properties Editor, World Properties** click on **Use Nodes.** In the Surface tab you will see that Surface is set to **Background.**

Click on the dot at the end of the **Color Bar** (Figure 15.33) and select **Environmental Texture** (background displays purple). In the menu that displays (Figure 15.34). Click on **Open** and navigate to your HDRI file in the Viewer.

Figure 15.34





HDRI Images

Image courtyard_night_4K.hdr

At this point you manipulate the view in the 3D Viewport Editor and move the Objects to position in the Scene as shown in Figure 15.37.

Figure 15.35
Figure 15.36

Have the 3D Viewport Editor with **Material Preview Viewport Shading** and with **Scene World checked** in the Shading options. **The** HDRI image will display as **the** Background to your Scene.

In Figure 15.37 the Objects have been positioned to appear as if they are sitting on the floor just **outside the open door. All the Lights in the Scene have been deleted showing that the Light from** the image affects the Objects.

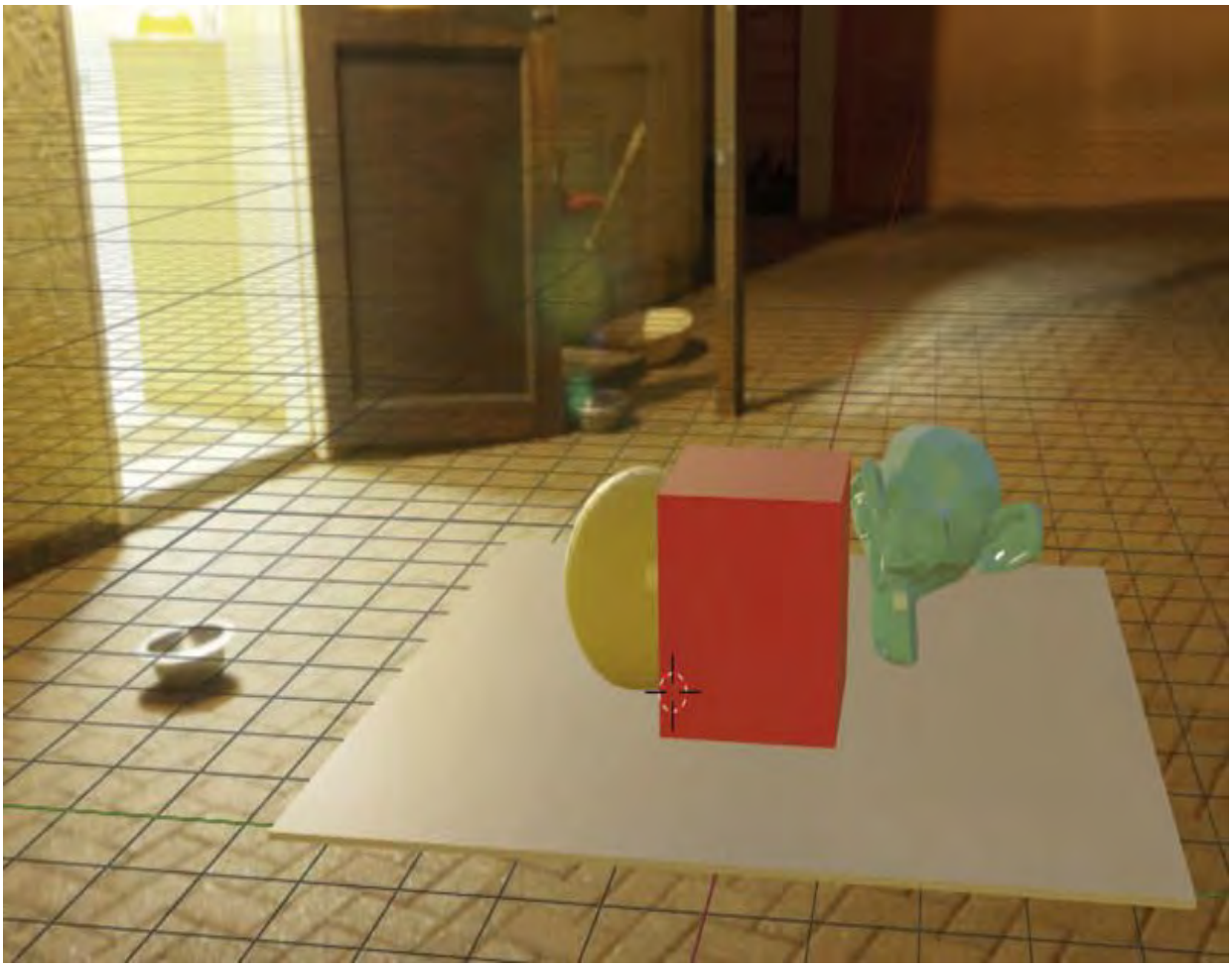


Figure 15.37

Images for Tracing and as Scene Backgrounds

HDRI Images will provide Scene Lighting but it may be that, simply placing an Image in the Scene to act as backdrop will do the job. Images may also be used as a template for Modeling.

Images for Tracing

To use an Image as a template for creating models you add an image file into the 3D Viewport Editor as a **Reference Image.** Be aware that in using this method the Image does **NOT Render.** Reference images are typically used for modeling. For example a front and side view of a human head (mugshot). To us an image in this way have the 3D Viewport in Front or Right Orthographic View. To enter an Image click **Add** in the Header and select, **Image** then **Reference** (Figure 15.38).

In **Blender File View** that opens navigate to the image you wish to use and click **Load Reference Image** in the lower RHS of the panel.

Object Mode
View
Select
Add
Object
Mesh
Curve
Surface
Metaball
Text
Grease Pencil
Armature
Lattice
Empty
Image
Light
Light Probe
Camera
Speaker
Force Field
Collection Instance
Reference
Background
Scene Collection
Collection
Camera
Cube
Empty
B_Head.jpg
Light
In the **Outliner Editor** you will see that the image has been entered in the 3D Viewport Editor as an **Empty Object** with the image attached.

Figure 15.38

On mouse over in the 3D Viewport Editor and the Empty (image) may be Translated, Rotated and Scaled but remember an Empty Object does **NOT** Render even though it displays in Rendered Viewport Shading Mode. As such, the image is suitable as a tracing template only.

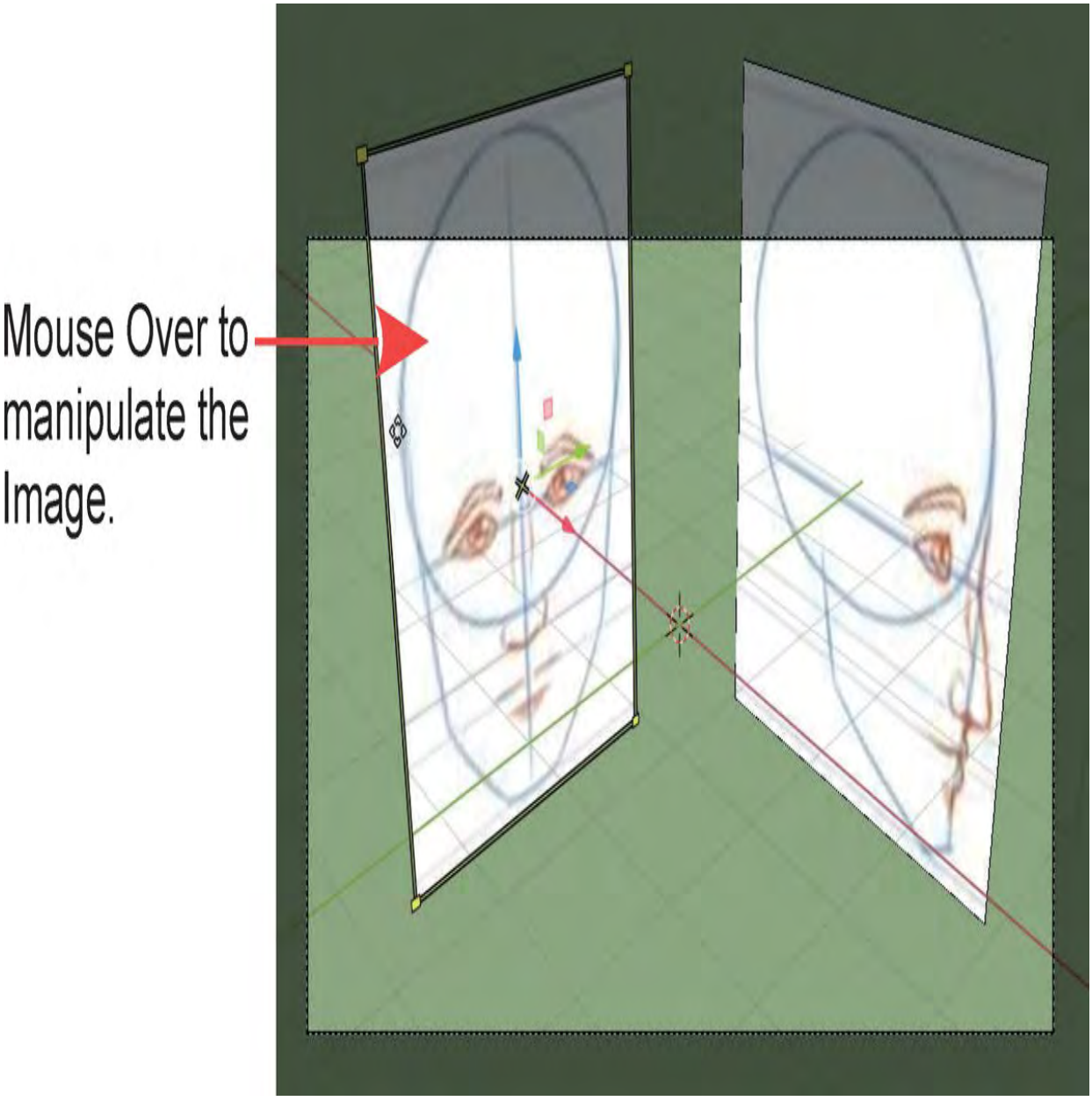
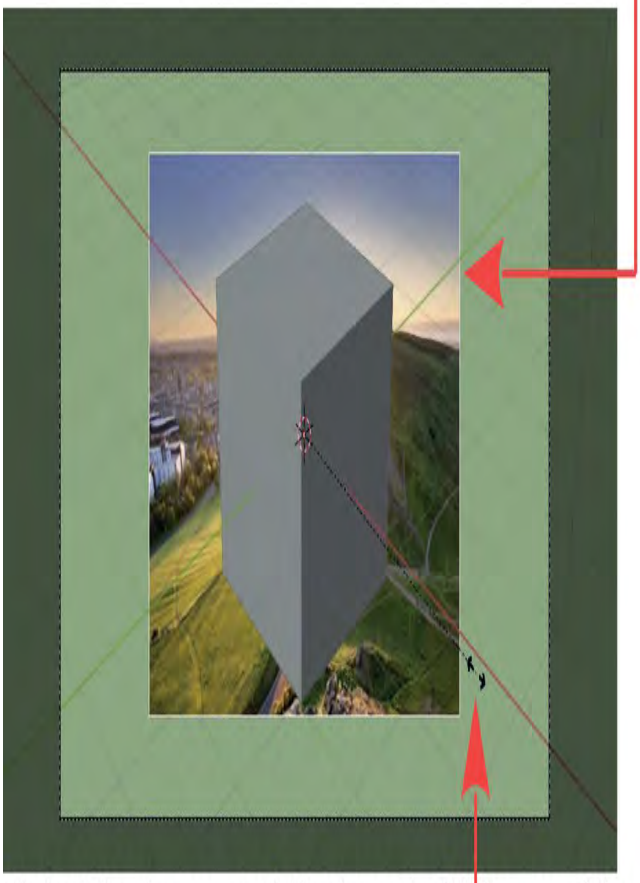


Figure 15.39

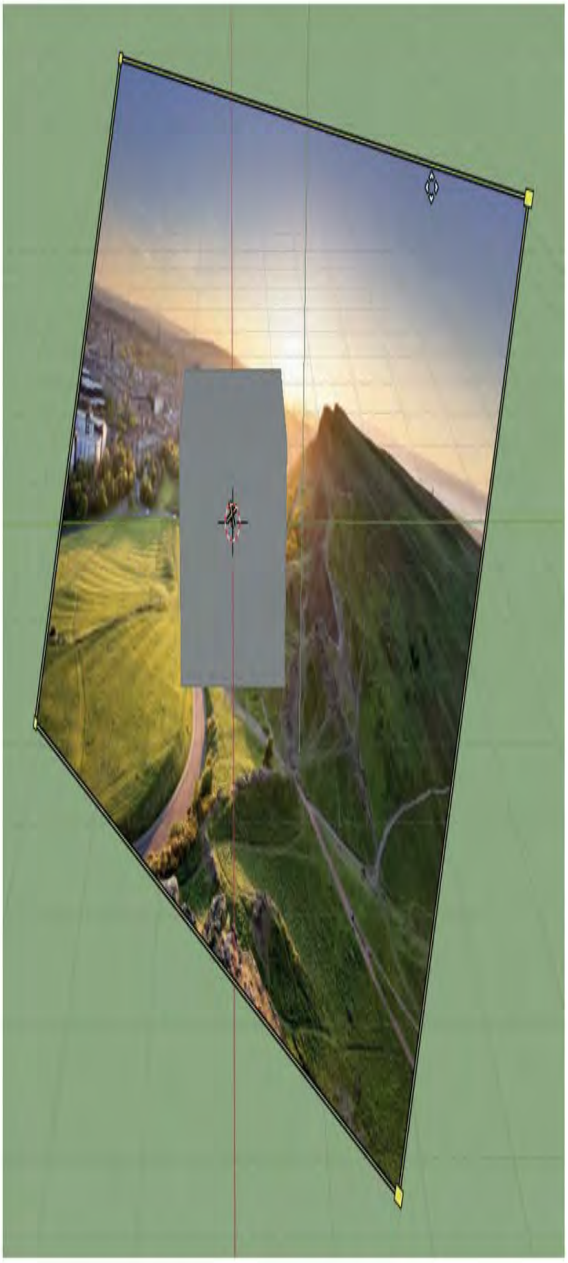
Reference Images positioned for Modeling. Camera View.

Image as a Scene Background

To introduce an image which Renders as a background to the Scene, be in Camera View and repeat the previous selection procedure, this time, selecting **Background**. Find your Background Image in Blender File View and click **Load Background Image**.



Select the Image and Scale up to fill Camera View.



BUT it does NOT Render.

Figure 15.40

Rotating the Viewport shows the Image positioned behind the default Cube (Figure 15.40).

Note: Entering the Image as Background allows you to see the Image in all Viewport Shading Modes **BUT it does NOT Render.**

To Render an Image as a background you have to Add the Image as a Plane and to do this you have to activate the **Import-Export Images as Planes** Add-on in the Preferences Editor. With the Add-on activated you will have the option to **Add Images as Planes** from the Add Menu.

Entering the Image in Camera View shows the Image (Plane) aligned along the Y Axis of the Scene at a reduced Scale (Figure 15.42). The Image (Plane) is located at the center of the Scene. You are required to Scale and Rotate the Image to fit the Camera View and move the Image back in the Scene behind any Objects. When the Image is entered as a Plane it is , in fact, an Object. Positioning the Image in the Scene, to be captured in Camera View, and relative to Scene Lighting has to be considered. You can have shadows being cast on the Image.

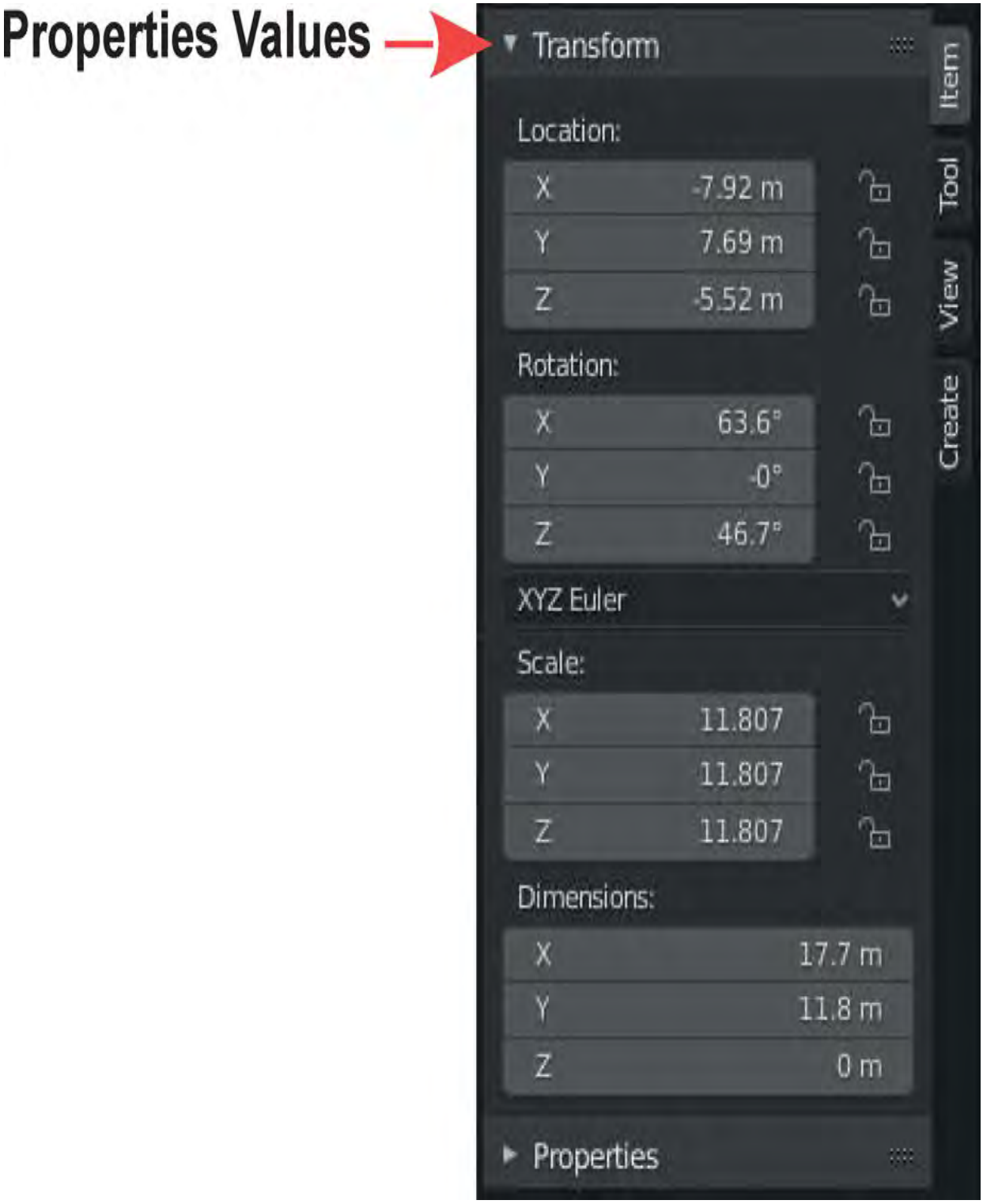

Properties Values
Transform
Location:
X -7.92 m
Y 7.69 m
Z -5.52 m
Rotation:
X 63.6°
Y -0°
Z 46.7°
XYZ Euler
Scale:
X 11.807
Y 11.807
Z 11.807
Dimensions:
X 17.7 m
Y 11.8 m
Z 0 m
Properties
Item
Tool
View
Create

Figure 15.41

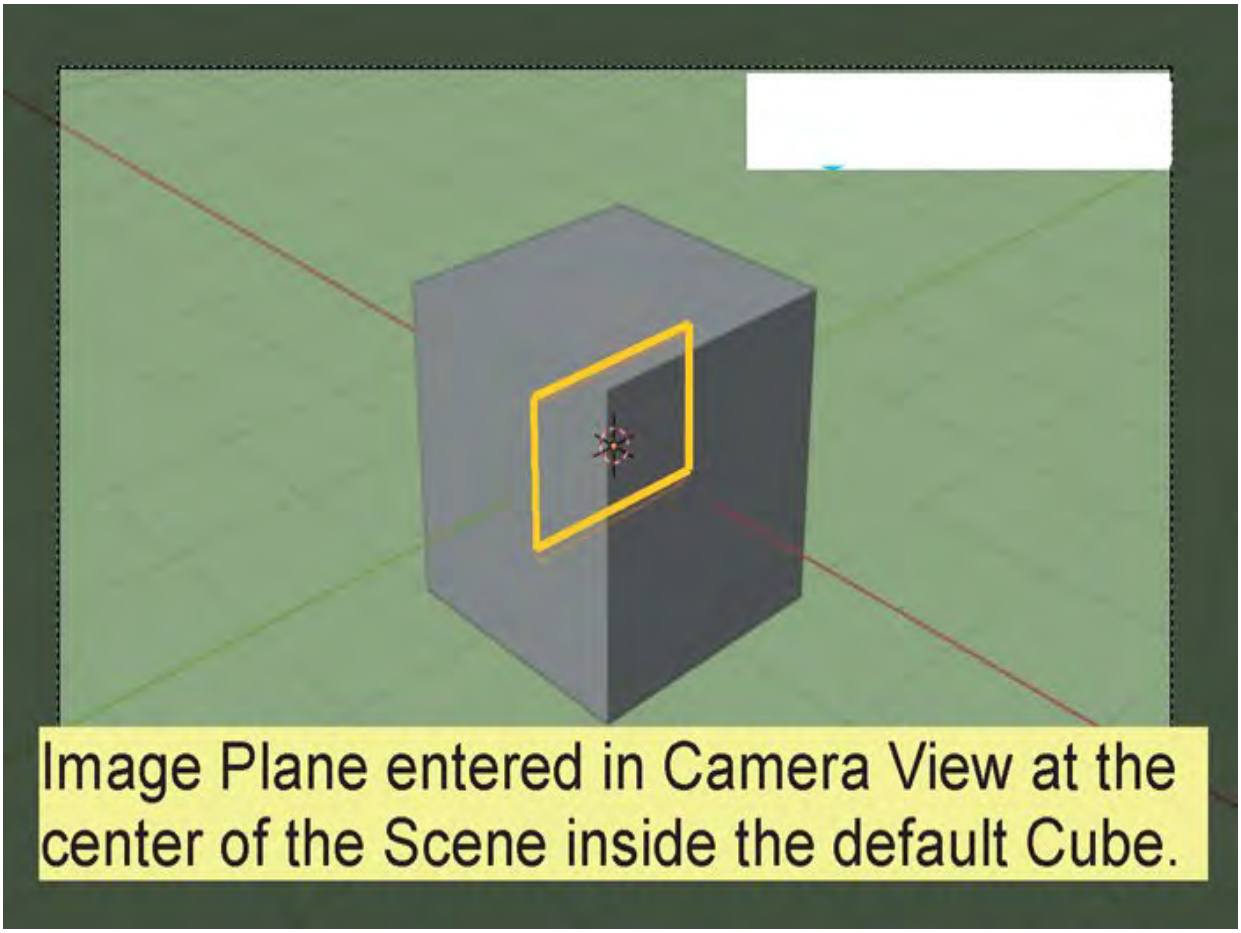


Figure 15.42

One method of positioning and Scaling the Image Plane is to be in Camera View with the Image Plane selected and have the Object Properties Panel opened (press the N Key). By adjusting values in the Properties Panel you will see the Image Plane fit the Camera View.

Camera View (Rendered Viewport Shading) with Properties Values adjusted.

Figure 15.43

Viewport Rotated showing the Image Plane located behind the default Cube.

Figure 15.44

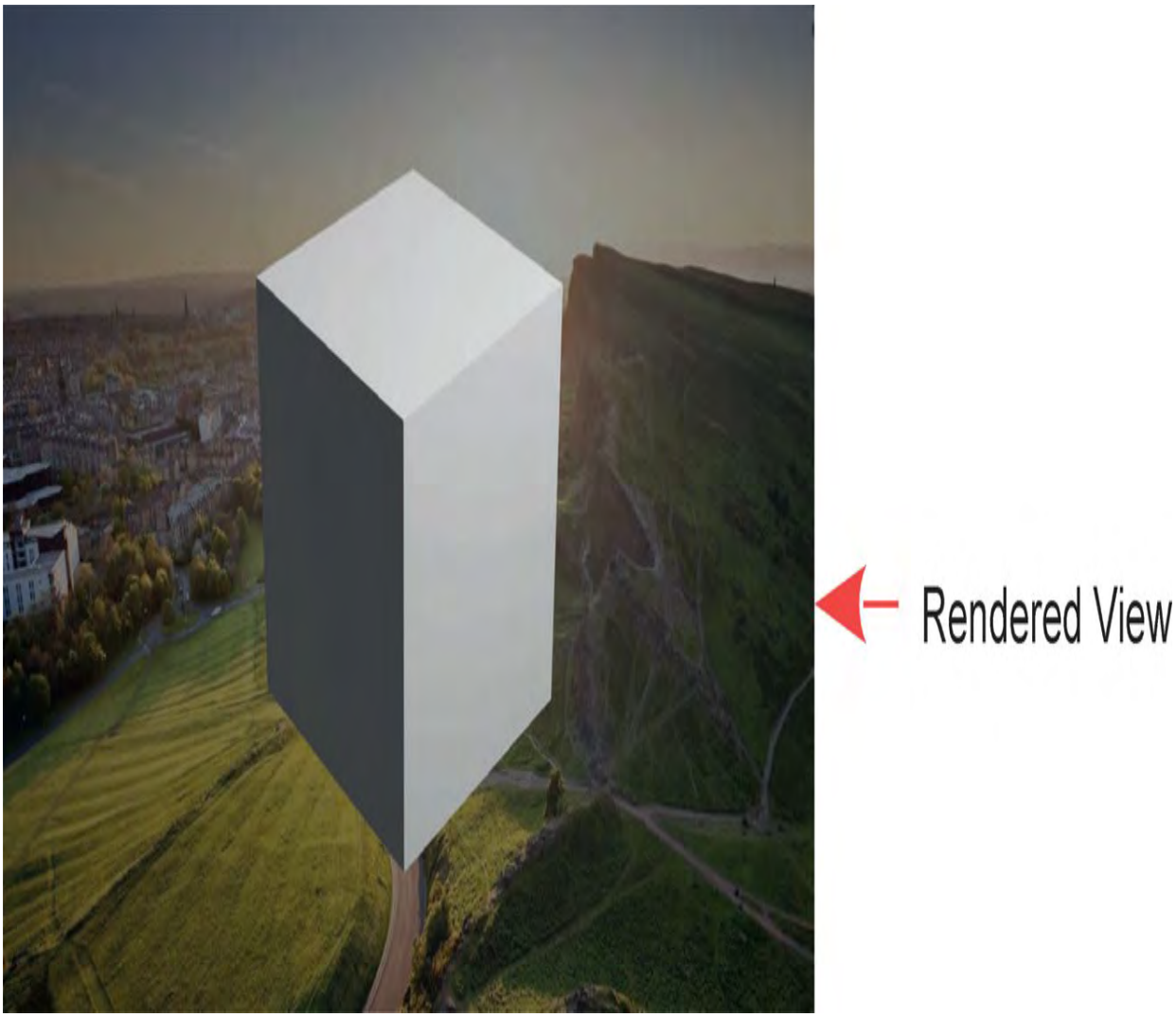


Figure 15.45

15.10 Volumetric Lighting

Using HDRI Maps as a background can give you a quick way to illuminate a Scene and combine a background but you are limited to what you can source in an image. You may want something completely unique and maybe not as complicated. Volumetric Lighting may be the answer.

Volumetric Lighting uses the light provided by Blenders Lights and scatters or diffuses the light in the Scene. A simple demonstration is to place a light source in a Scene behind an Object such that the diffuse light shines through the Object, casting shadows towards the Camera.

To produce a Volumetric Lighting effect as shown in Figure 15.46 Blender's Node System is used. How to use Nodes is explained in Chapter 16 – 16.11. Volumetric Lighting is explained in Chapter 18.

With the Eevee Render System Volumetric Lighting is also used to create a Mist or Fog effect.

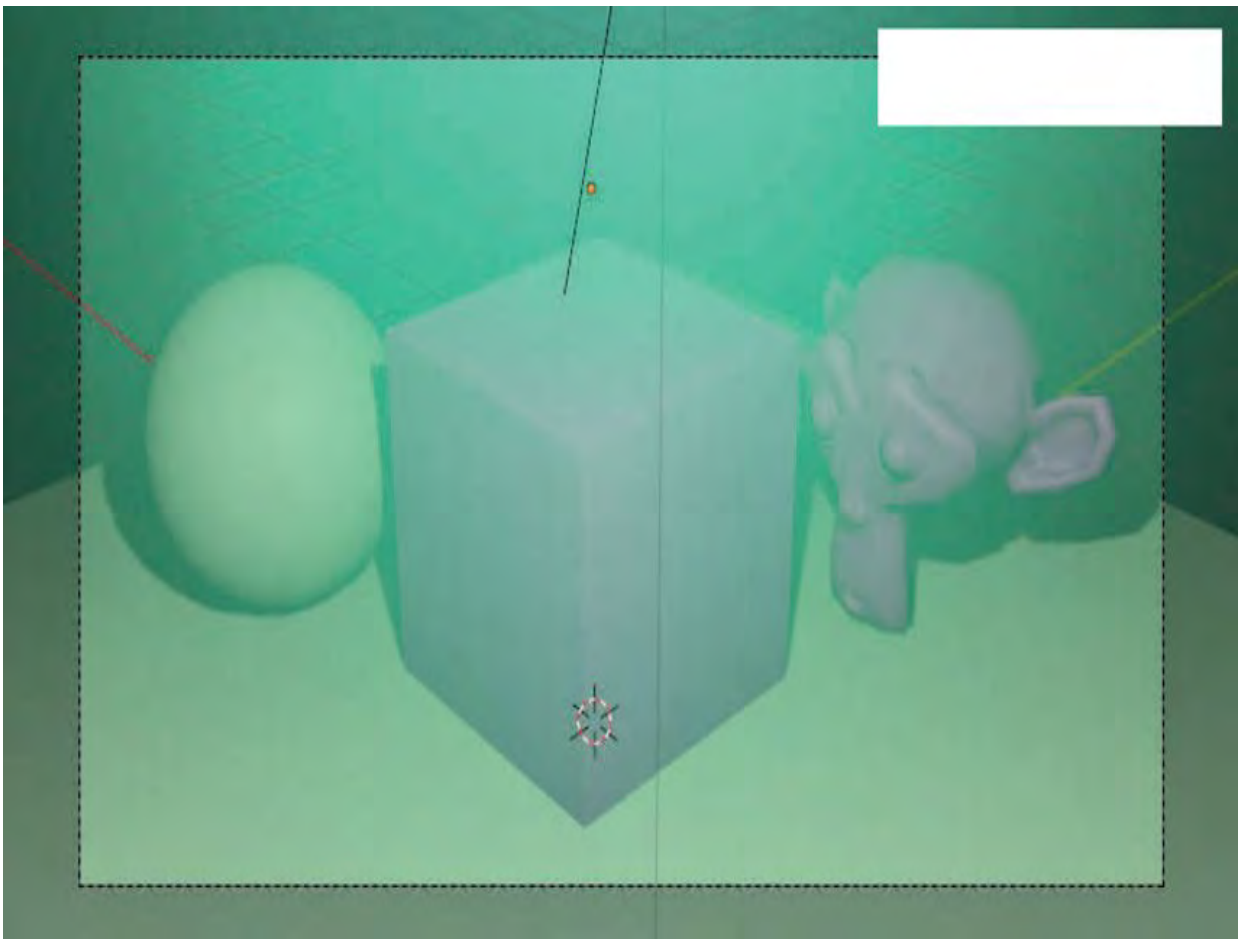


Figure 15.46

16 Materials Texturis Nodes

16.1 Material Definition
16.2 Materials in Practise
16.3 Default Cube Material Properties
16.4 Adding Material Properties
16.5 Alternative Material Assignment
16.6 Multiple Material Datablocks
16.7 The Material Slot
16.8 The Color Picker
16.9 Material Display
16.10 Lighting Affects
16.11 Materials Using Nodes
16.12 What Are Nodes
16.13 Accessing and Viewing Node Effects
16.14 Noodle Curving
16.15 The Shader Editor
16.16 The Shader Node Menu
16.17 Adding Nodes
16.18 The Shading Workspace
16.19 Scene Arrangements
16.20 Mixing Material Example
16.21 The Principled BSDF Node
16.22 The Color Ramp Node
16.23 Saving Screen Space
16.24 Vertex Paint

16.1 Materiai Definition

In Computer Graphics, how the surface of an Object displays is determined by three factors; **Material, Texture and Lighting.**

Material: The Base Color of the surface.

Texture: The physical characteristics of the surface.

Lighting: The background illumination or light emitting from Lights (Lamps).

Material may be considered as the color of an Object which is how the visible spectrum of light reflects from the Object's surface. A Material also defines whether the surface appears dull (matt) or shiny (metallic).

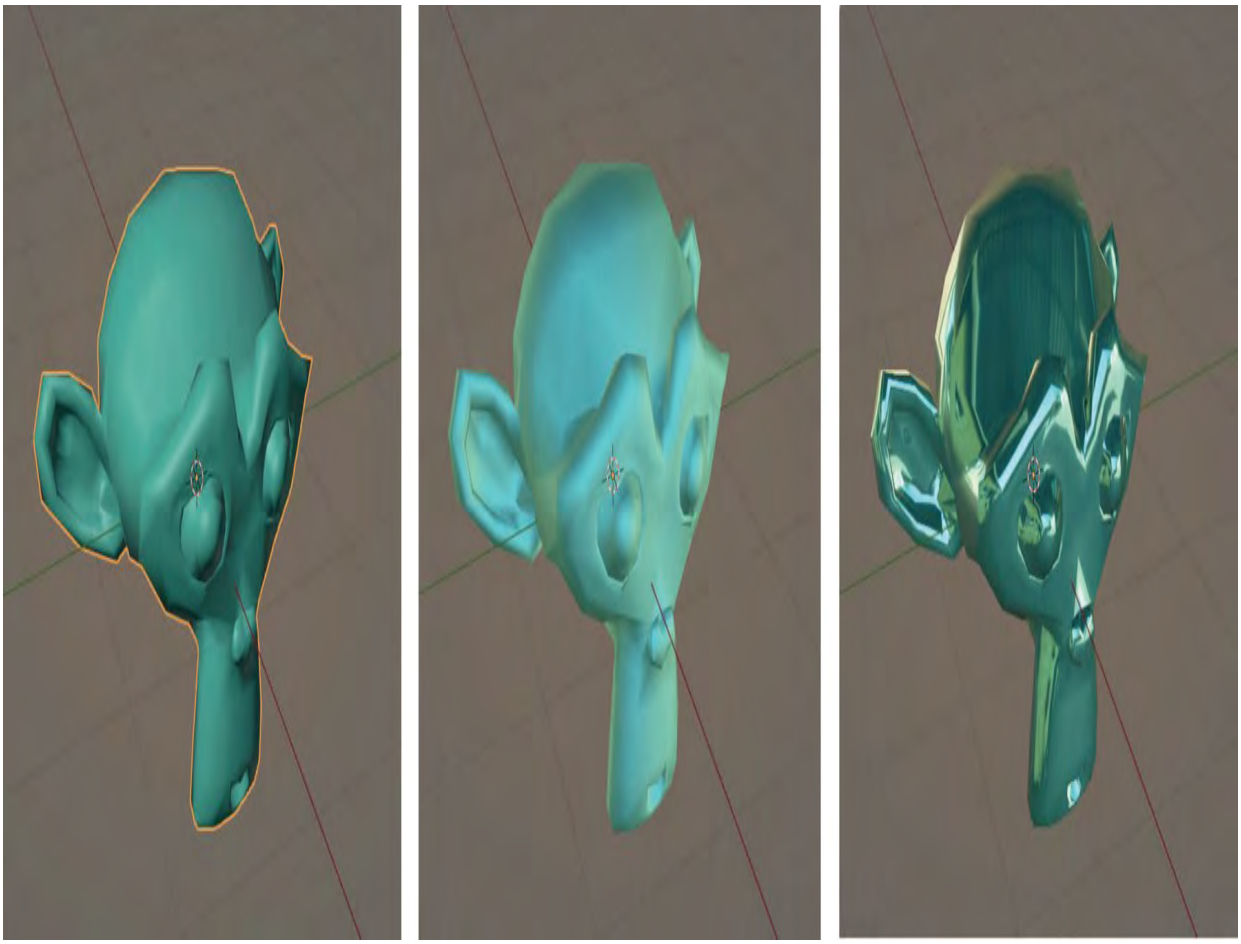


Figure 16.1

16.2 Materials in Practise

Materials are said to be, **Applied to an Object.** In its simplest form this concept means coloring an Object. In Chapter 1-1.12 it was demonstrated how to color the Monkey Object (Suzanne) by disabling the default **Node System** and adjusting controls in the Properties Editor. Before expanding on the subject you should take a step back and understand what is taking place in the default 3D Viewport Editor when Blender is first opened.

The default Scene contains a **Cube Object** which displays with a gray color (Figure 16.2 below). The 3D Viewport Editor is in **Solid Viewport Shading Mode** (Reference Chapter 14 — 14.3). With the Cube Object selected, the **Properties Editor, Material Properties** display controls for the **Material** (color) with Blender's **Material Node System** engaged. This is indicated by **Use Nodes** being highlighted blue (Figure 16.4). The Material color of the Cube is set by the **Base Color** in the Properties Editor. Clicking the Base Color bar displays the **Color Picker.** The **RGB** values: 0.800 and Alpha: 1.000 produce the default gray.

Nodes are a graphical representation of code which is configured to produce the display in the 3D Viewport Editor. Nodes are used to control many features but in this instance the Nodes control the Material Color. By changing the 3D Viewport Editor to the **Shader Editor** (Reference: Chapter 2 — 2.1) you see a Node named: **Principled BSDF** connected to a Node named: **Material Output** (Figure 16.3). The values set in the Principled BSDF Node are displaying the gray color on the surface of the Cube Object. The Node is specific to the selected Object (Cube).

You will observe that the controls in the Principled BSDF Node replicate the controls in the Properties Editor, Material Properties, Surface Tab (Figures 16.3 — Figure 16.4).

Clicking the **Base Color Bar** in the Shader Editor, Principled BSDF Node or in the Properties Editor, Material Properties, Surface Tab display the color

picker for selecting and changing the color (Figure 16.4).

Note: In the default Scene the Material Properties are set with the **Node System active** as indicated by **Use Nodes** being highlighted blue (for Nodes see 16.12).

Note: The 3D Viewport Editor is in **Solid Viewport Shading Mode** (Reference Chapter 14).

When attempting to change the Base Color of the Cube, **with the Nodes Active,** you will not see a color change in the 3D Viewport Editor. You have to change the Viewport Shading to **Material Preview** or **Rendered** Viewport Shading Mode. Deactivate the Nodes by clicking the **Use Nodes** button to see the color change in Solid Viewport Shading Mode.

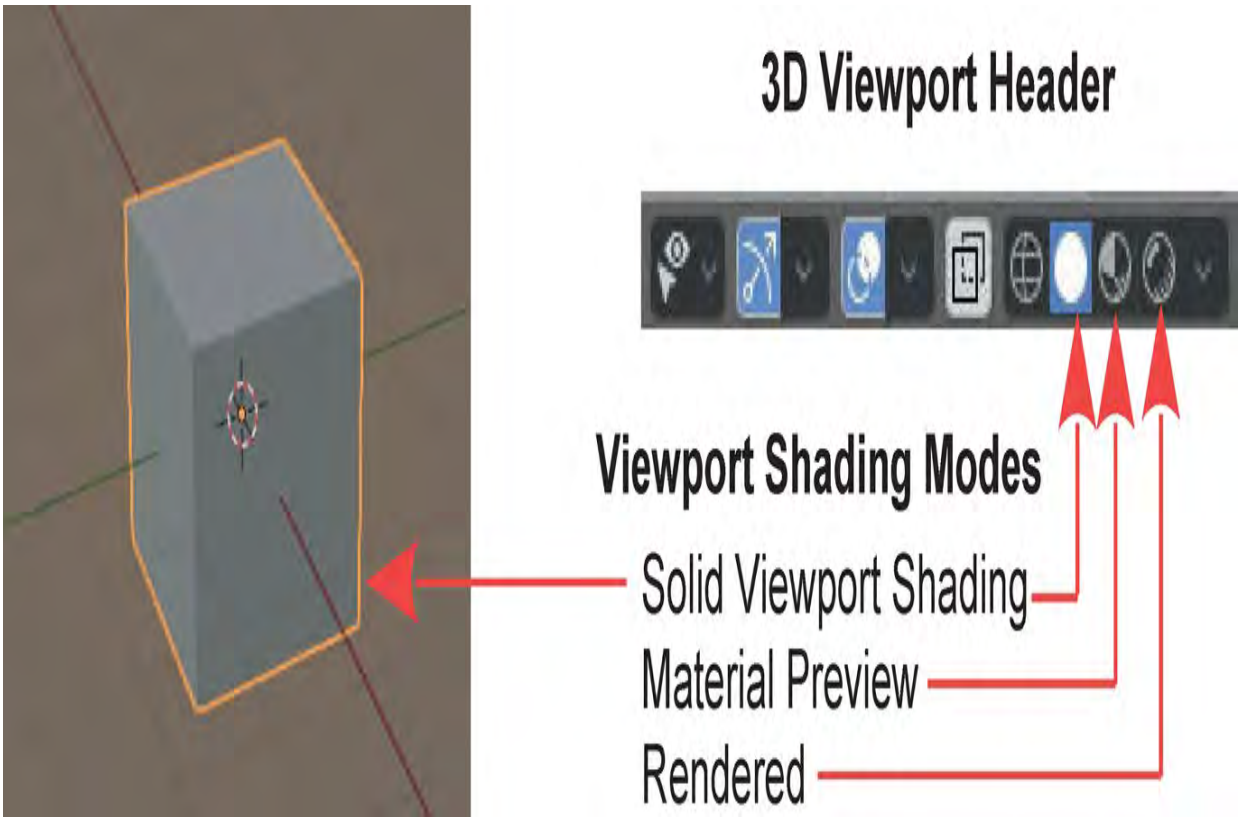


Figure 16.2

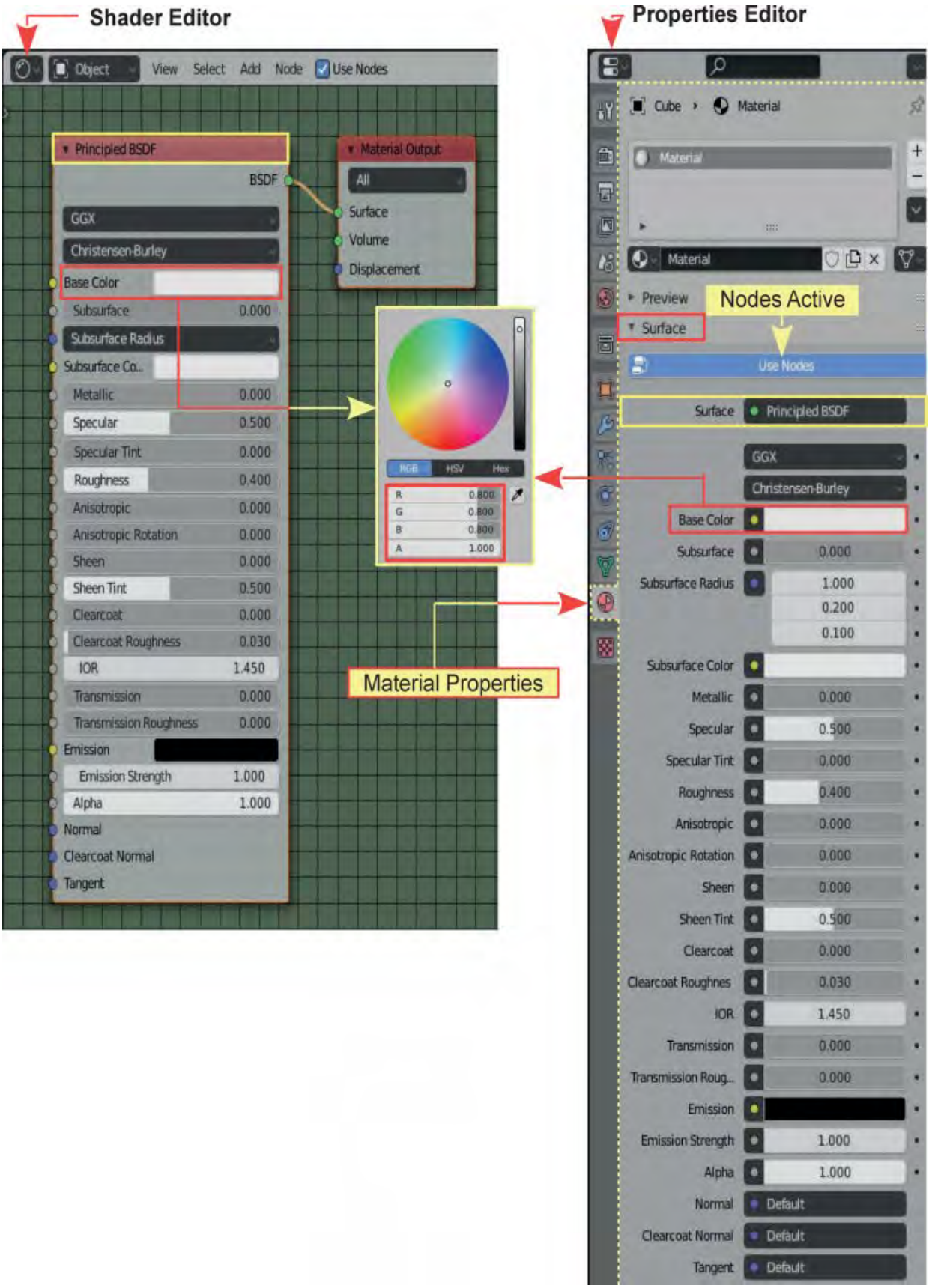

Shader Editor
Properties Editor
Object
View
Select
Add
Node
Use Nodes
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius
Subsurface Co...
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.400
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement
RGB
HSV
Hex
R 0.800
G 0.800
B 0.800
A 1.000
Material Properties
Cube
Material
Material
Preview
Surface
Nodes Active
Use Nodes
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius 1.000 0.200 0.100
Subsurface Color
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.400
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughnes 0.030
IOR 1.450
Transmission 0.000
Transmission Roug... 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal Default
Clearcoat Normal Default
Tangent Default

Figure 16.3

Figure 16.4

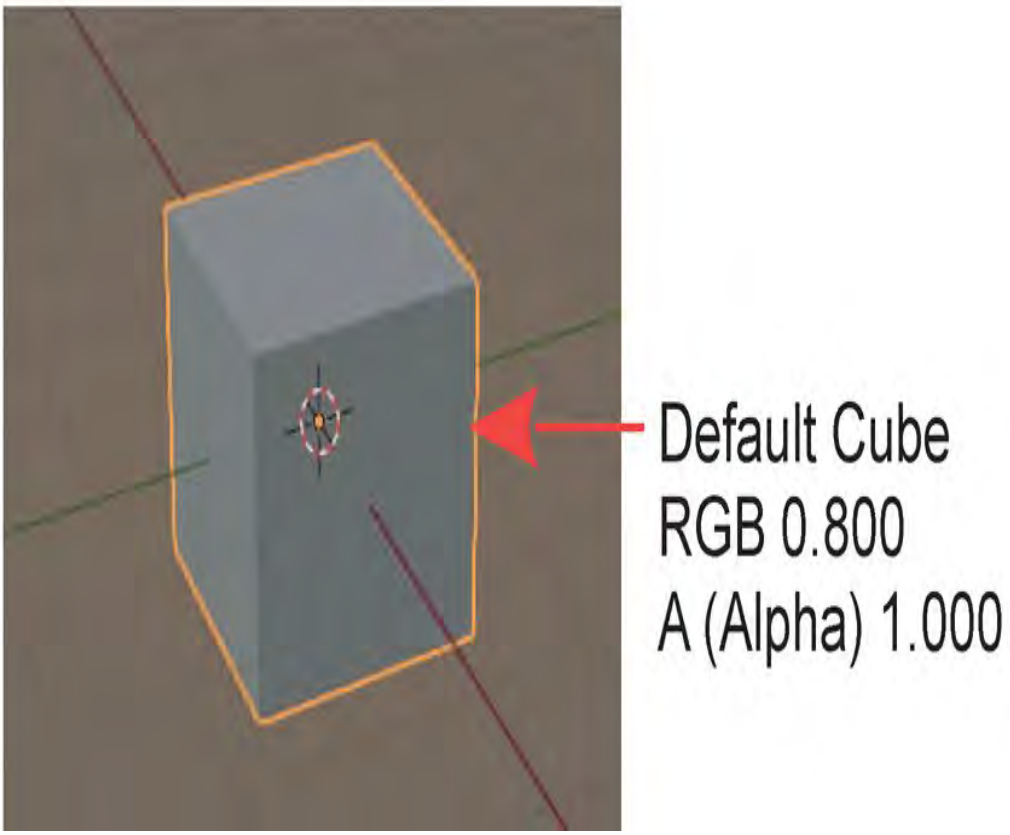


Figure 16.5

Nodes Disabled (Inactive)

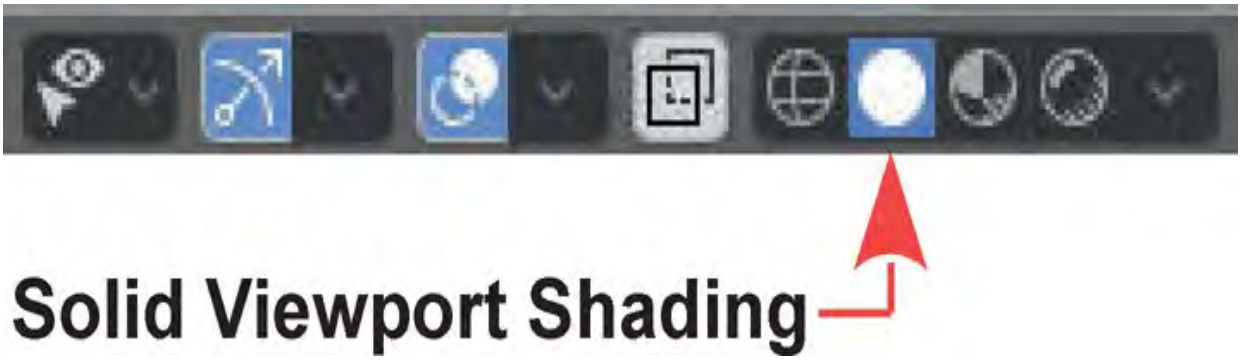


Figure 16.6

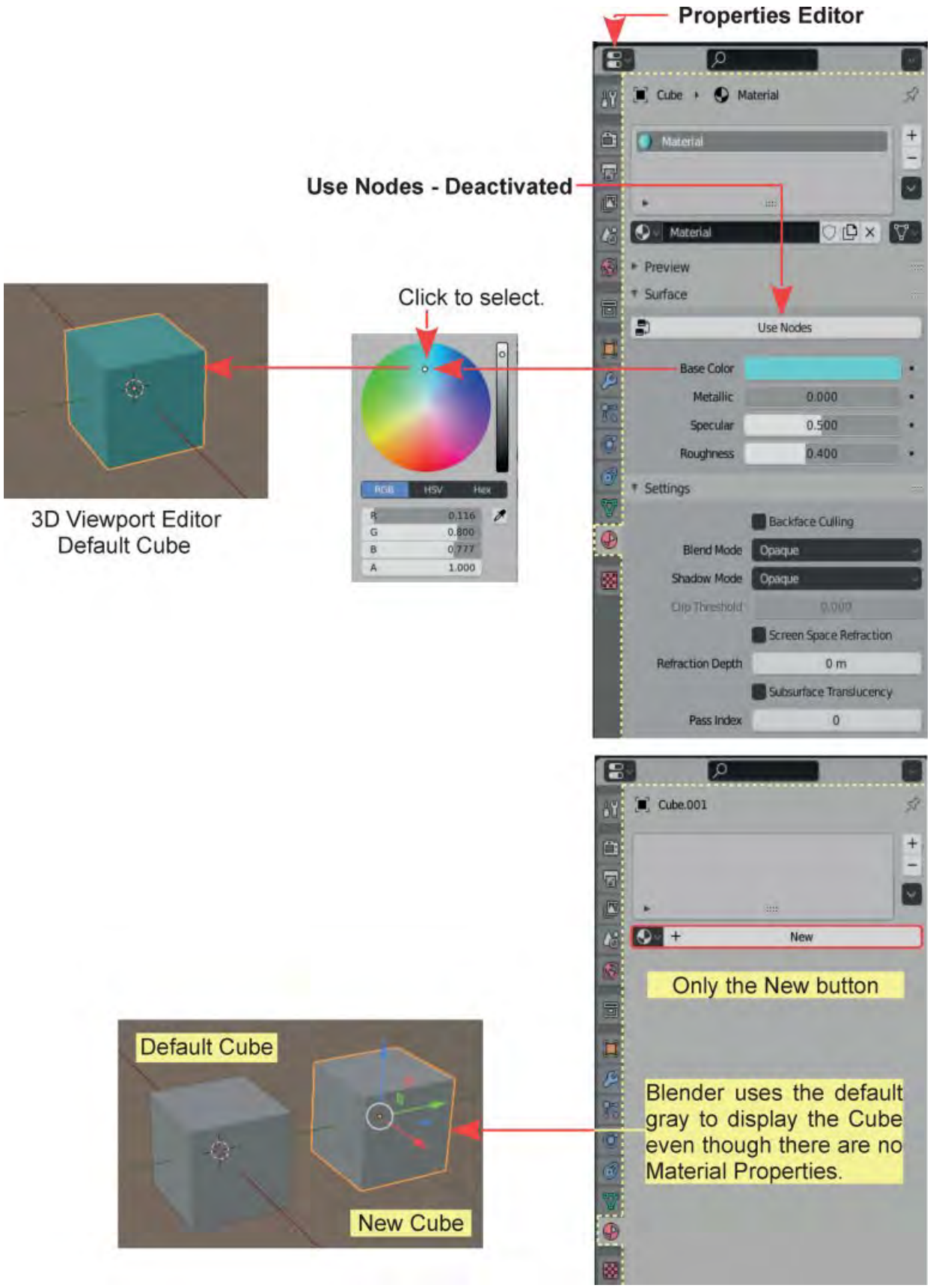

Properties Editor
Use Nodes - Deactivated
Click to select.
3D Viewport Editor
Default Cube
Cube
Material
Material
Material
Preview
Surface
Use Nodes
Base Color
Metallic
0.000
Specular
0.500
Roughness
0.400
Settings
Backface Culling
Blend Mode
Opaque
Shadow Mode
Opaque
Clip Threshold
0.009
Screen Space Refraction
Refraction Depth
0 m
Subsurface Translucency
Pass Index
0
RGB
HSV
Hex
R
0.116
G
0.800
B
0.777
A
1.000
Cube.001
New
Only the New button
Default Cube
New Cube
Blender uses the default gray to display the Cube even though there are no Material Properties.

Figure 16.7
Figure 16.8

With Use Nodes deactivated (gray) click the Base Color bar and select a color. The color is added to the Cube.

Note: The **Shader Editor** continues to display the Principled BSDF Node connected to the Material Output Node even though the Nodes are deactivated (Figure 16.3).

16.3 Default Cube Material Properties

When Blender is first opened displaying the default Cube Object in the default Scene, the Cube has Material Properties pre-applied. The Properties display in the Properties Editor, Material Properties Tab. **Note: This only occurs with the default Cube.**

When a new Cube Object or any of the Primitive Objects are entered into the Scene there is no Material applied. You see this in the Properties Editor, Material Properties. There is only a **New Button** showing (Figure 16.7).

16.4 Adding Material Properties

To add Material Properties to the New Cube or any New Object entered in the Scene you press the **New Button** in the Properties Editor, Material Properties. Pressing the New Button assigns a duplicate **set of data** which produces the default gray color. You adjust values in the Properties Editor to change the color.

The **set of data** is called a **Datablock** which for the default Material is simply named **Material.**

Before pressing the New Button, have the default Cube selected in the 3D Viewport Editor and examine the upper panel in the Properties Editor, Material Properties.

Clicking the **Browse Material to be Linked button** displays the **Material Cache** containing the single **Datablock named Material.**

Datablock: Block of Data (Code) producing the Material (Color).

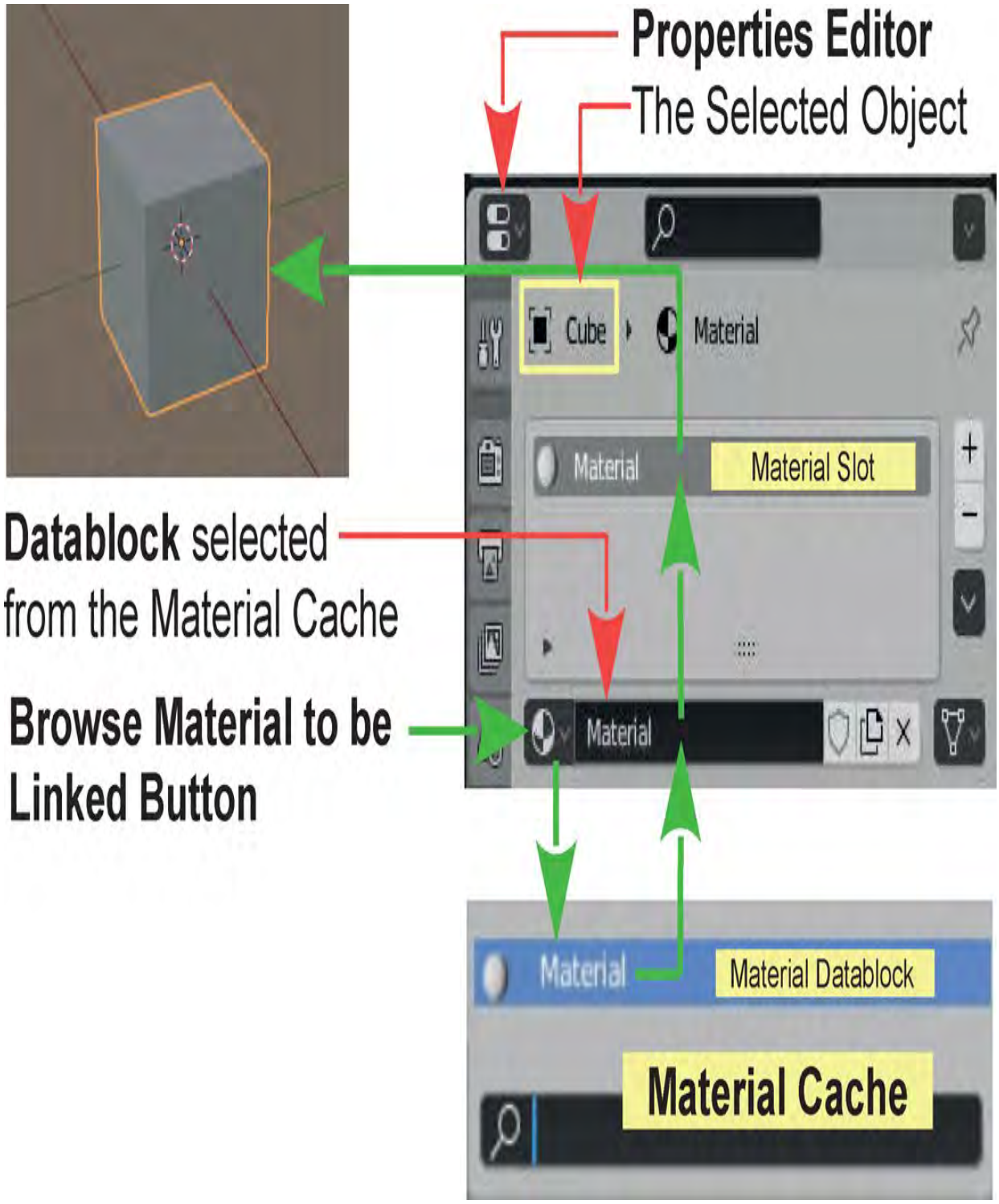


Figure 16.9

The **Datablock** has been selected from the **Cache** and entered into the **Material Slot** which applies the Datablock to the selected Object (the default

Cube) in the 3D Viewport Editor.

At this point there is only one **Material Datablock** in the **Cache,** the one named **Material.**

Enter a new Cube in the Scene.

Go ahead and press the **New Button.**

Click **Use Nodes** to cancel the Node System.

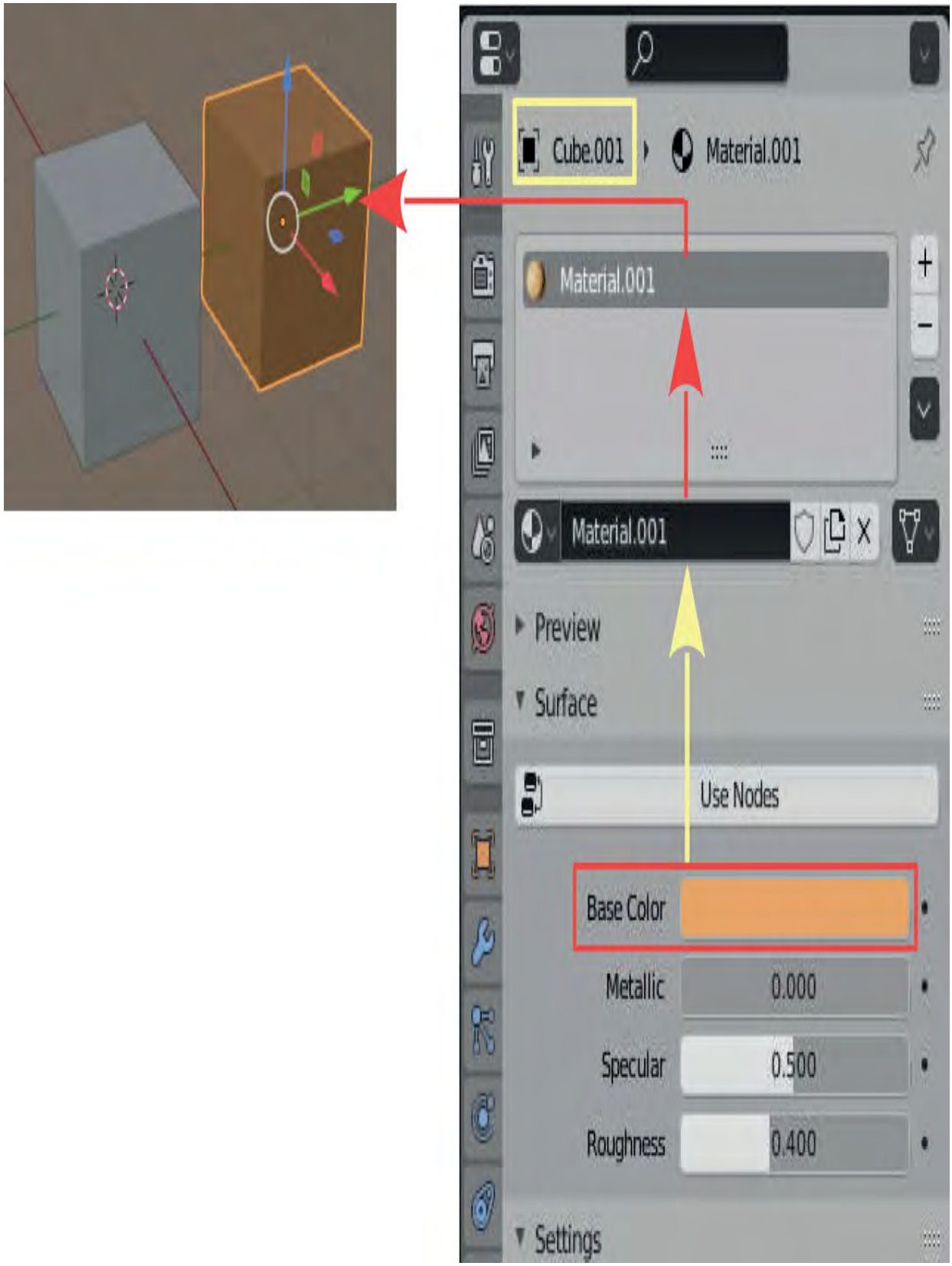

Cube.001
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Base Color
Metallic
0.000
Specular
0.500
Roughness
0.400
Settings

Figure 16.10

Pressing the New Button (with the **new Cube Object** selected) enters a new Material Datablock named **Material.001.**

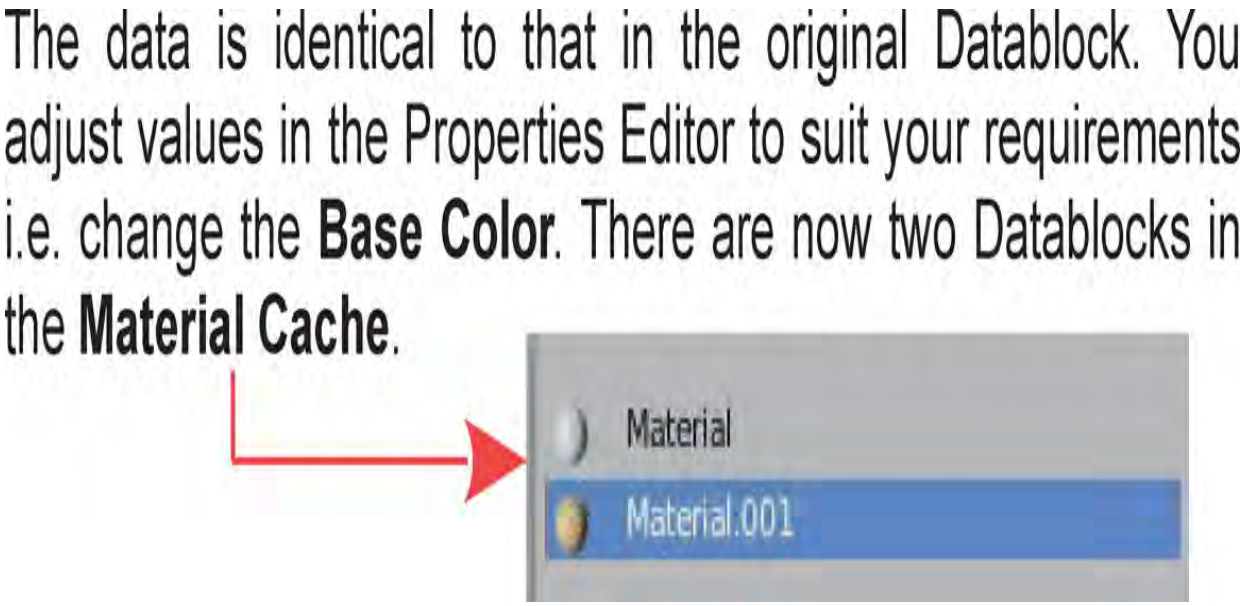


Figure 16.11

16.5 Alternative Material Assignment

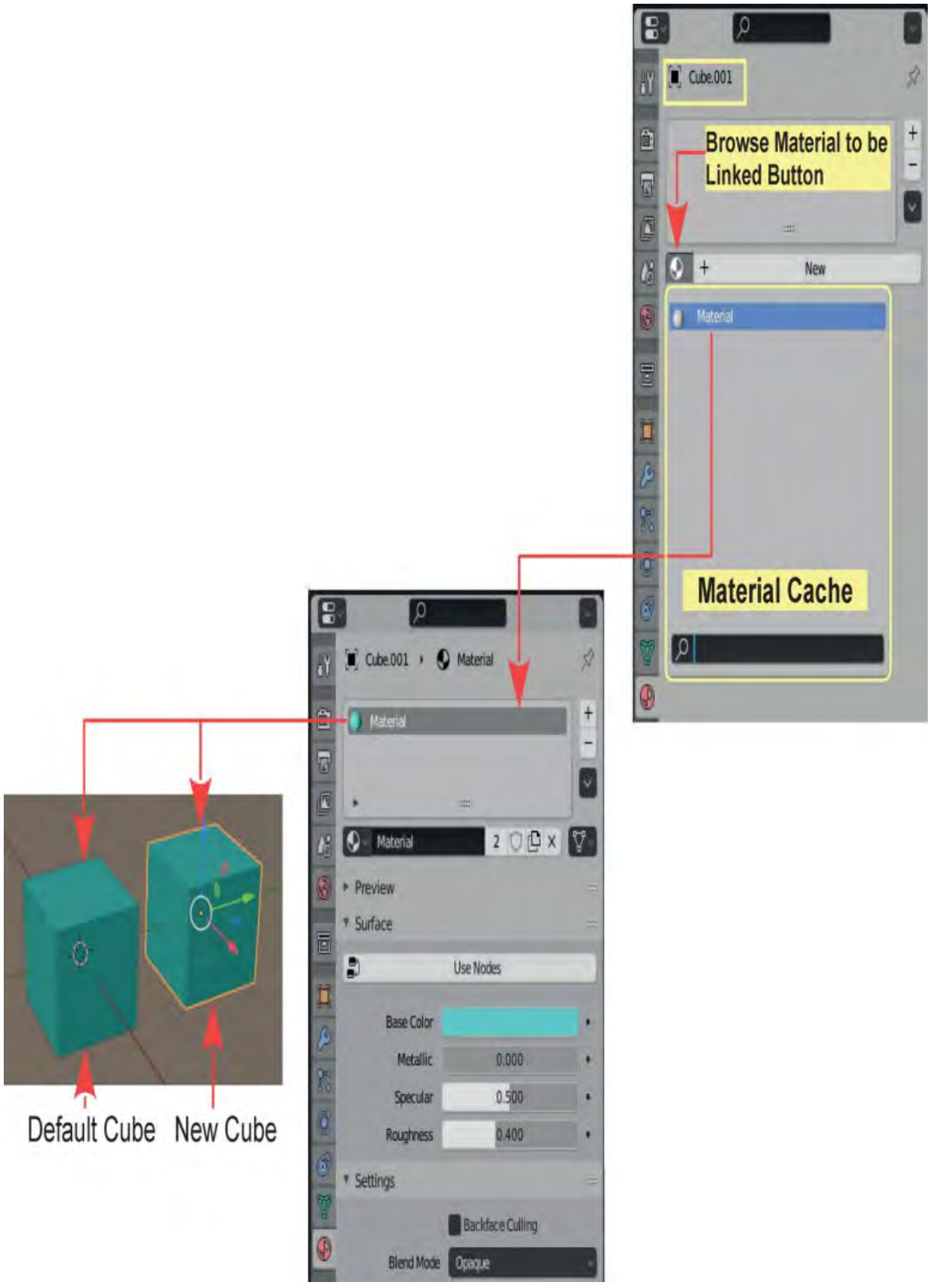

Cube.001
Browse Material to be Linked Button
New
Material
Material Cache
Cube.001 › Material
Material
Material
2
Preview
Surface
Use Nodes
Base Color
Metallic
0.000
Specular
0.500
Roughness
0.400
Settings
Backface Culling
Blend Mode
Opaque
Default Cube
New Cube

Figure 16.12

Figure 16.13

Note: If **Use Nodes** is activated the same will apply.

Note: there is only one Material Datablock in the Cache.

With a new Cube entered in the 3D Viewport Editor and selected, click the **Browse Material to be Linked** Button instead of the New Button then click on the Datablock named **Material.**

The Datablock named **Material** will be assigned to both the default Cube and the new Cube and Material Properties will display in the Properties Editor. Both Cubes will display gray in the 3D Viewport Editor.

Changing the **Base Color** with either Cube selected changes the color of both Cubes. Changing any value in the Properties Editor changes the value for both Cubes.

16.6 Multiple Material Datablock:

Assigning Material Properties by clicking the New Button when new Objects are added to the Scene generates new Material Datablocks in the Cache.

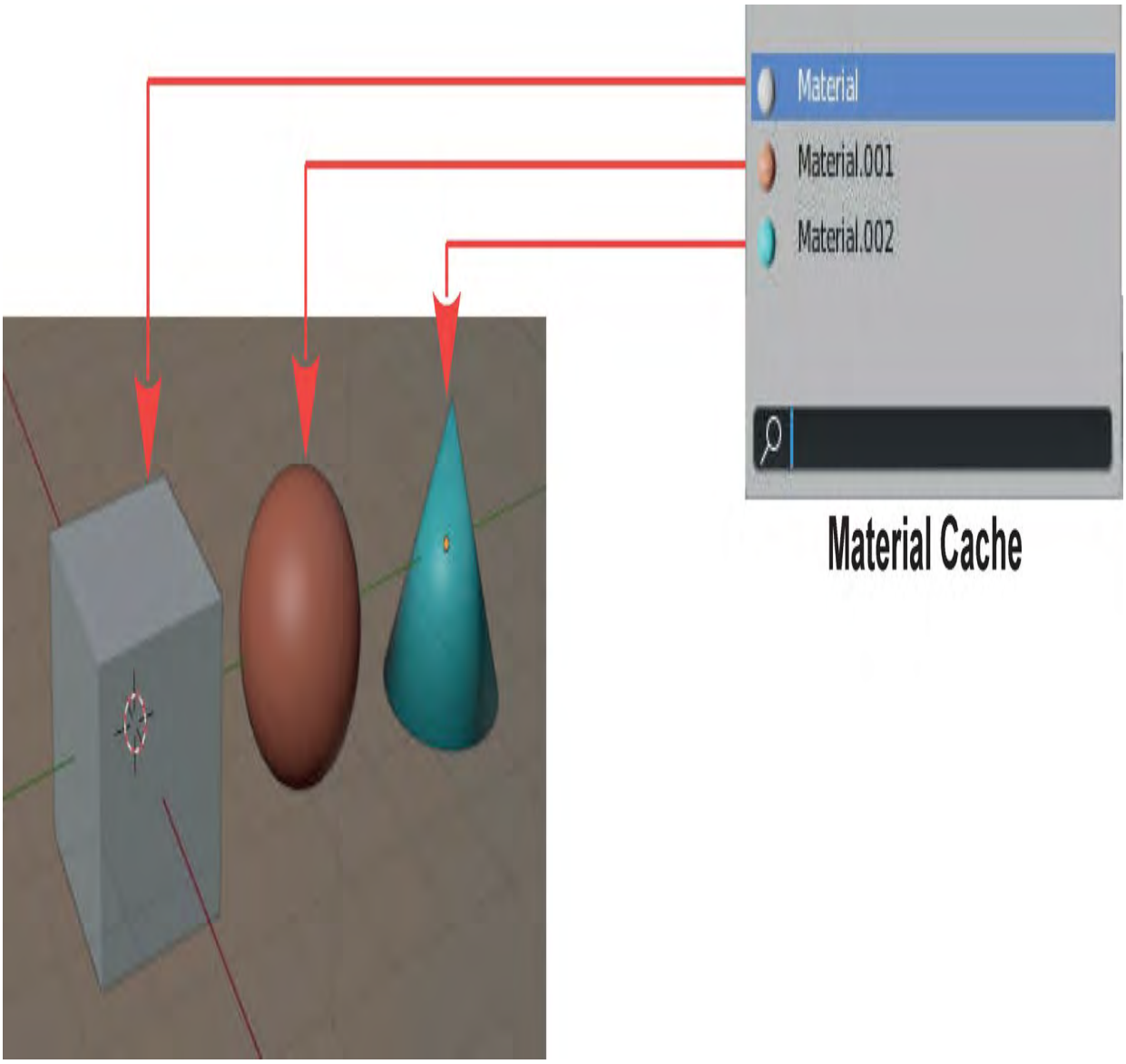


Figure 16.14

By selecting an Object in the 3D Viewport Editor, clicking the Browse Material to be Linked button, and clicking one of the Datablocks in the Cache, assigns that Material Datablock to the selected Object.

16.7 The Material Slot

Figure 16.13 shows a Material Datablock selected from the Cash and entered in the Material Slot the Datablock assigns the Material to the selected Object

in the 3D Viewport Editor.

Figure 16.14 demonstrates that with multiple Objects each with a different Material color there are multiple Datablocks available for selection.

Selecting a Datablock from the Cash automatically enters the Datablock in the Material Slot hence applying it to the selected Object.

Adding Material Slots

With an Object selected you may add Material Slots and apply multiple colors to an Object. One method of doing this is to create **Vertex Groups** (Reference Chapter 5 -5.9) and apply a Material to a group.

To demonstrate, open a new Blender File, delete the default Cube and enter a UV Sphere into the Scene and give it a new Material (Material.001 Figure 16.15).

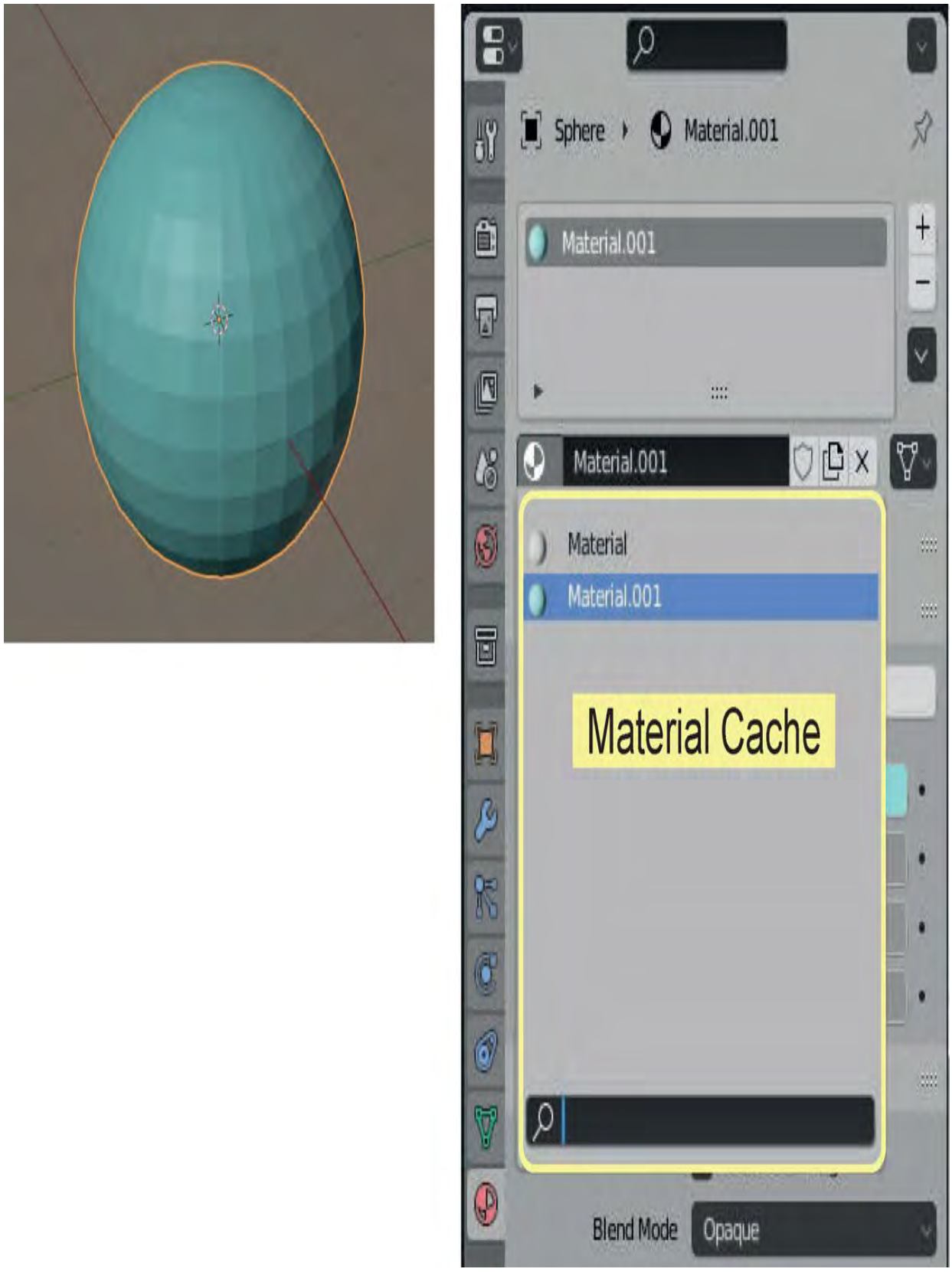

Sphere
Material.001
Material.001
Material.001
Material
Material.001
Material Cache
Blend Mode
Opaque

Figure 16.15

Note: When giving the UV Sphere a New Material it is automatically named **Material.001.** In the Cache you will see **Material** and **Material.001.**

Material is the default Material used on the deleted Cube Object

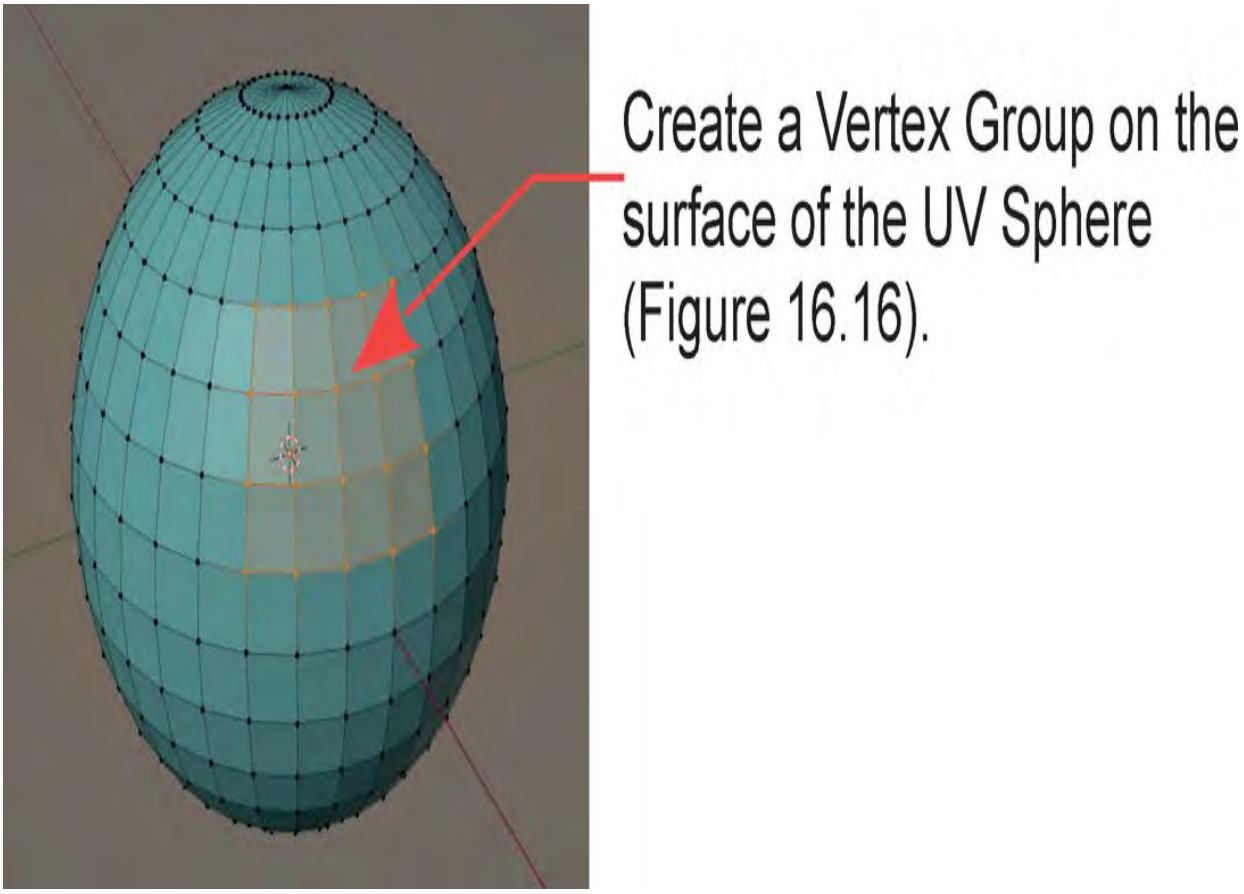


Figure 16.16

Remember: If you are working in a Blender File where you have entered several Objects and have given them different Materials (Reference: Figure 16.14) and then deleted the Objects, the Materials that were applied to the Objects will remain in the Material Cache.

With the UV Sphere in question, it will be assumed you have started a new Blender File, deleted the default Cube and entered a new UV Sphere. In this case the Cache will show the default Material named **Material** and the Material named **Material.001** (Figure 16.15).

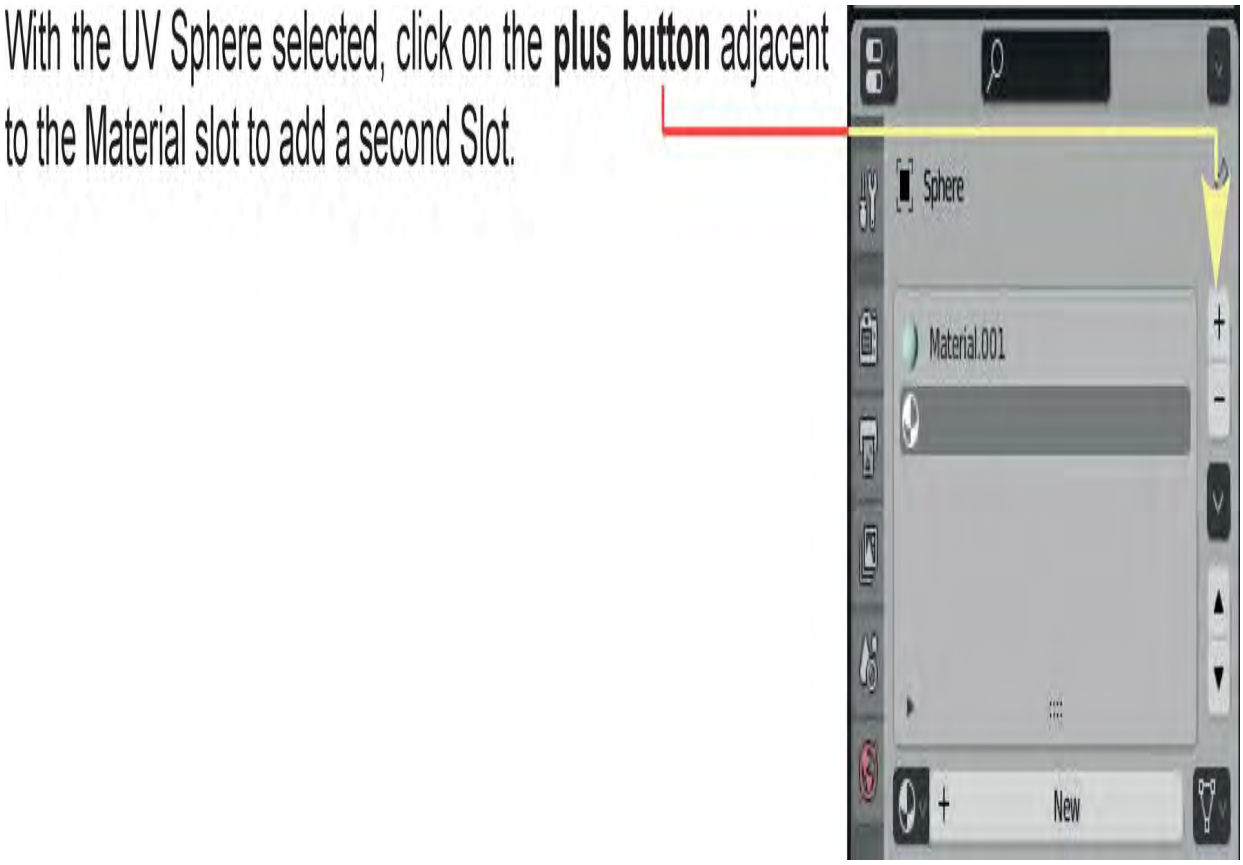


Figure 16.17

You may open the Cache and select a Material to enter in the new Slot or alternatively, click the New button, cancel the Node System and select a new Base Color to create a new Material.

The new Material is entered in the new Material Slot and is automatically named **Material.002** (Figure16.18).

The UV Sphere does not change color since Material.001 is being applied from the original Slot.

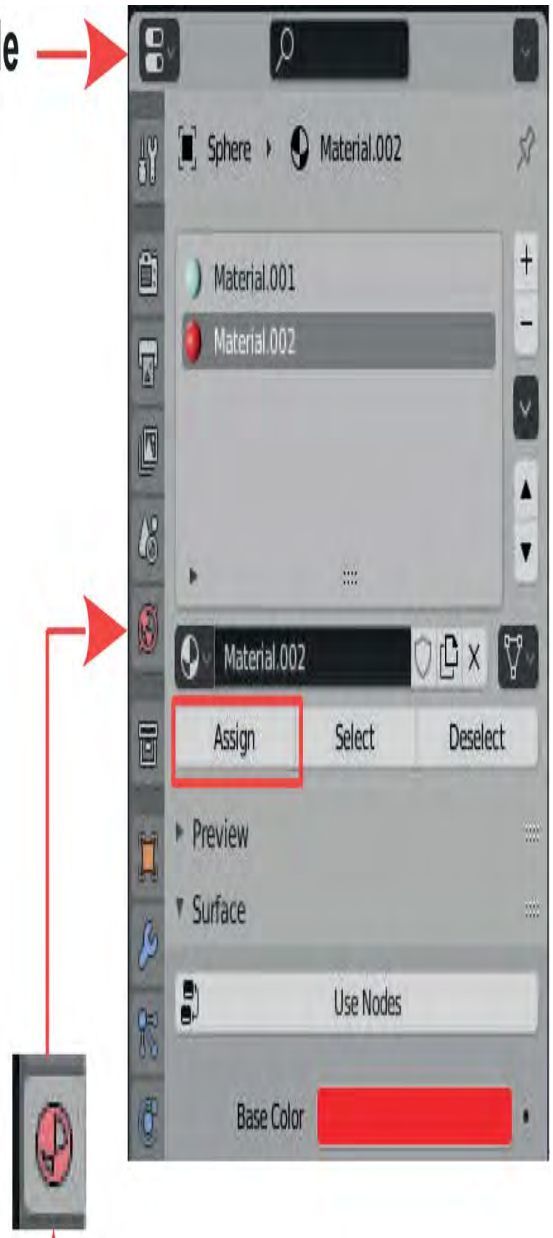
Figure 16.18

Have the UV Sphere with the Vertex Group created in **Edit Mode** in the 3D Viewport Editor.

In the Properties Editor, Material Properties with the new slot selected (highlighted) containing Material.002, click the **Assign** Button (Figure 16.19).

Material.002 is displayed on the Vertex Group in the 3D Viewport Editor (Figure 16.20 — 16.21).

Properties Editor, Material Properties, Edit Mode



Note: Figures 16.17, 16.18 and 16.19 show the Properties Editor with the Material Properties selected.

Figure 16.19

Note: The Assign button only displays with the 3D Viewport Editor in Edit Mode.

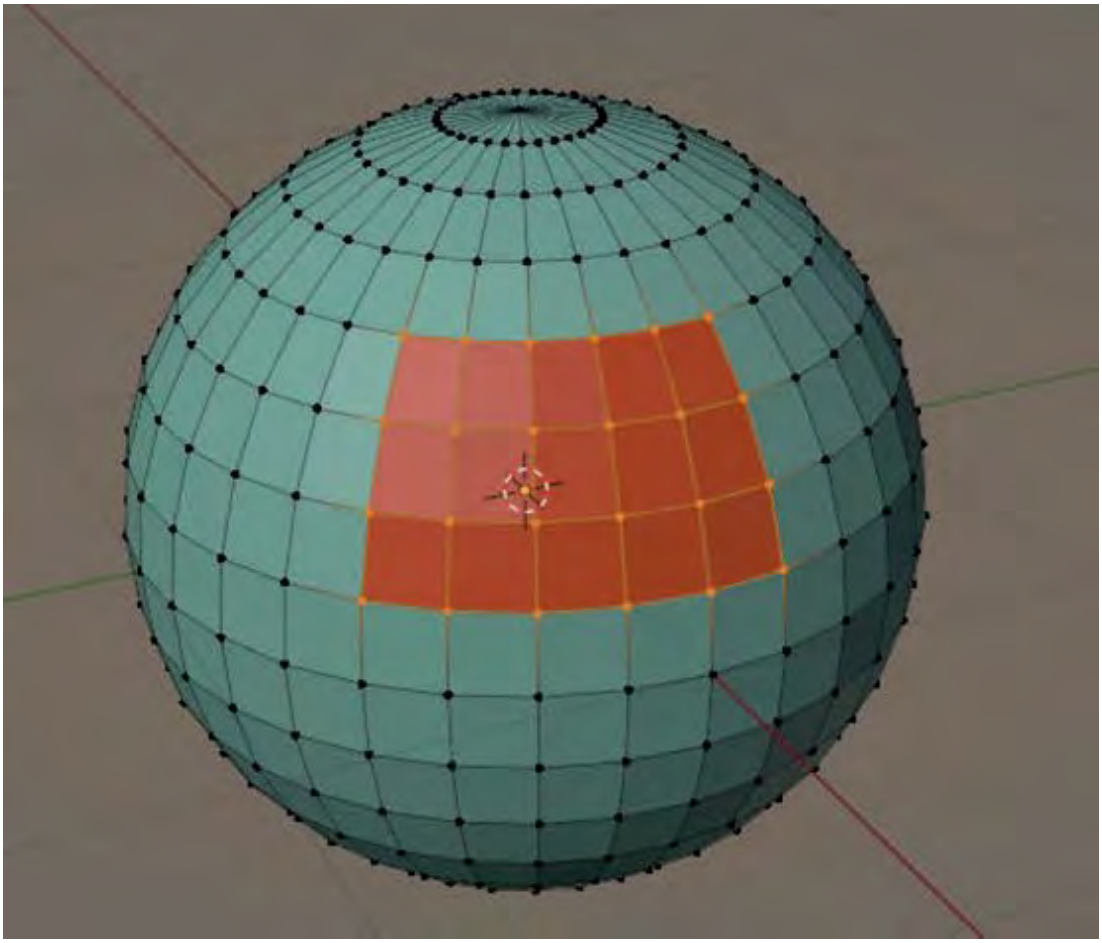


Figure 16.20

Edit Mode

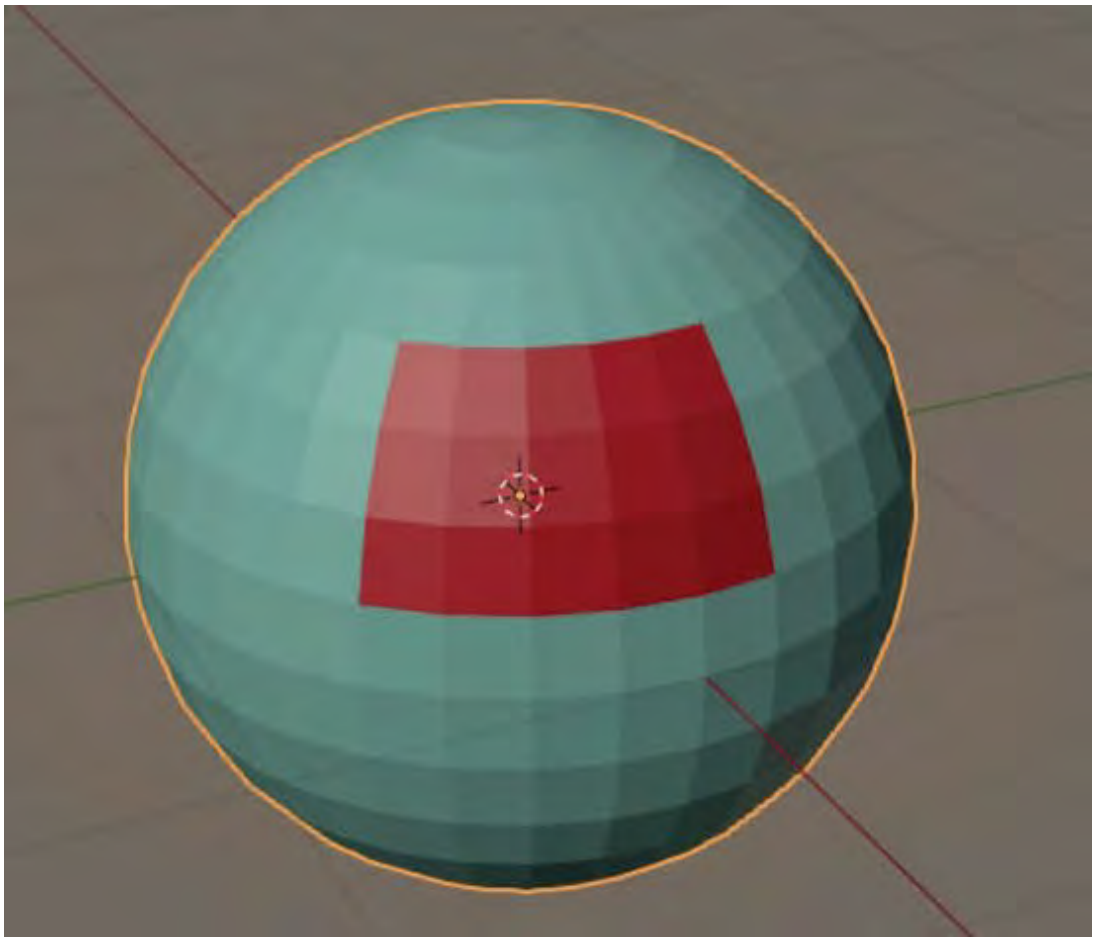


Figure 16.21

Object Mode

16.8 The Color Picker

The Color Picker is activated from a variety of locations in the interface such as clicking the Base Color bar in the Properties Editor, Material Properties.

In practise, a Material (color) is selected in the Color Picker Circle.

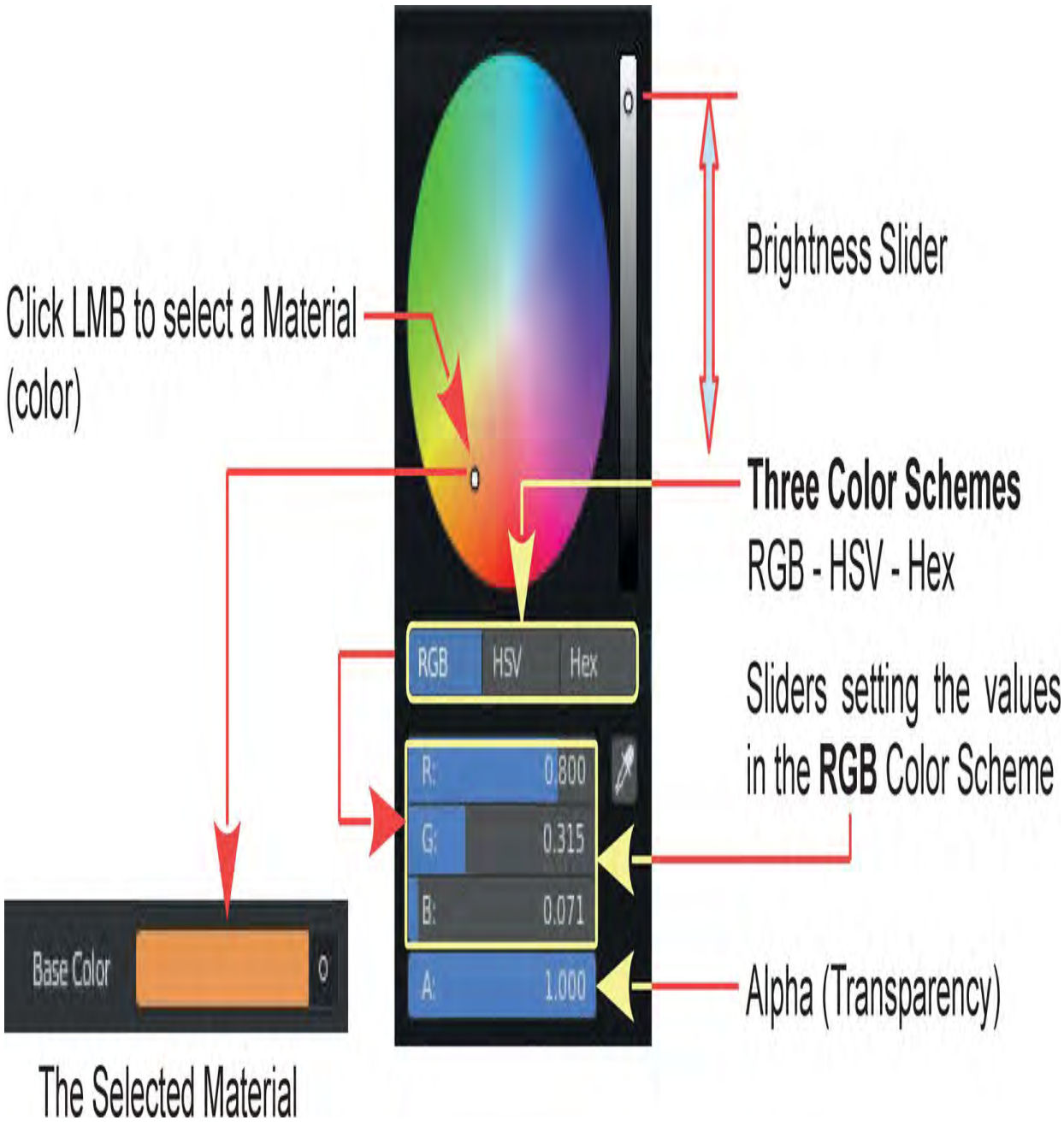


Figure 16.22

Materials (color) can be displayed in three color schemes i.e. **RGB, HSV** or **Hex.** How the color appears in each scheme is also dependent on an Alpha value which is the amount of transparency the material (color) is given and the brightness of the color. (Chapter 1, 1.12 — Coloring).

16.9 Material Display

In the foregoing Material (color) has been applied to the selected Object in the 3D Viewport Editor having the Node System deactivated. With the Node System inactive you see the Material in all Viewport Shading Modes. Up to this point, apart from the Base Color that has been selected, how you see the Materials has been determined by the **default settings** in the Properties Editor, Material Properties.

How you see, the Material may be controlled by adjusting the values in the Properties Editor. To demonstrate, have a Monkey Object, shaded smooth with the following settings.

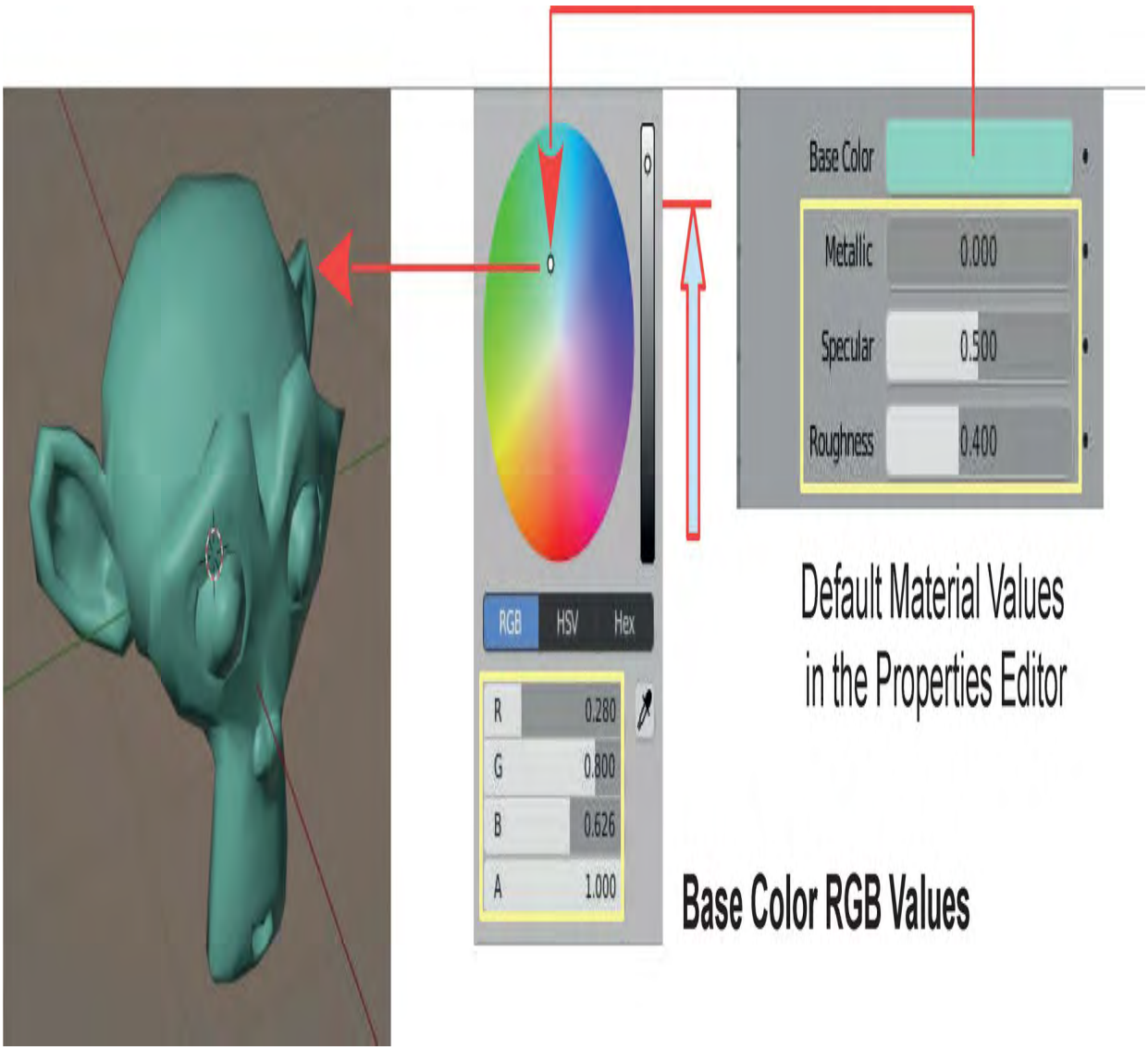
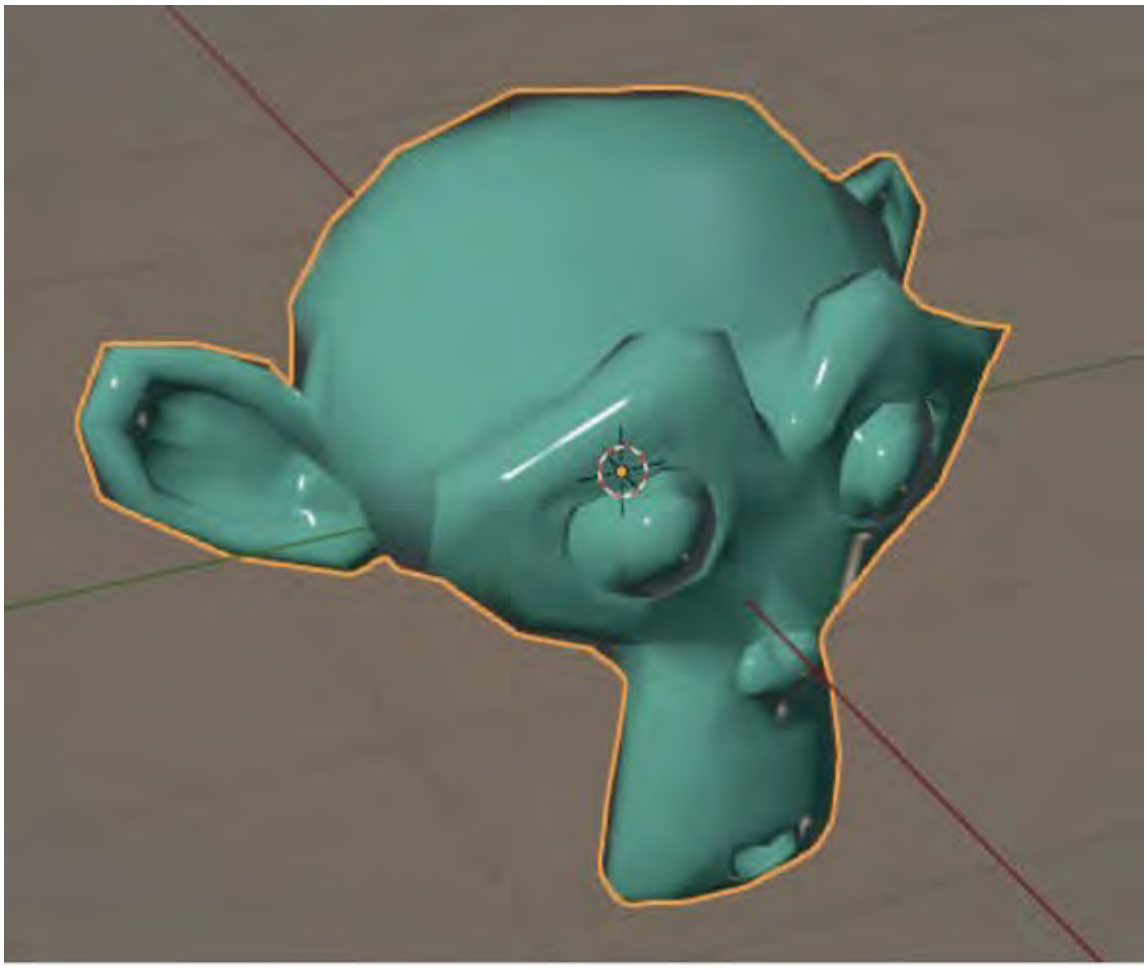


Figure 16.23

Base Color RGB Values

Adjust values in the Properties Editor, Material Properties to see the difference.



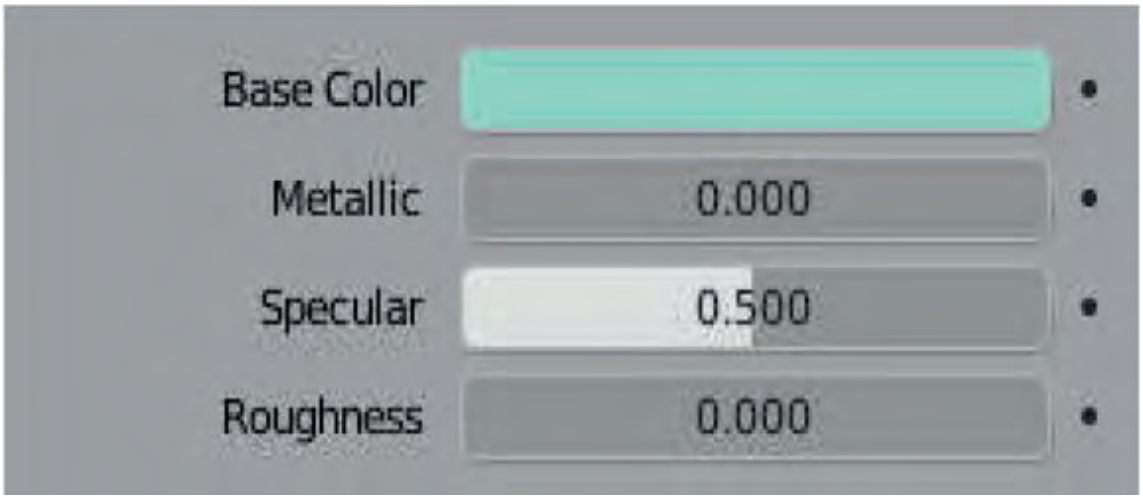


Figure 16.24

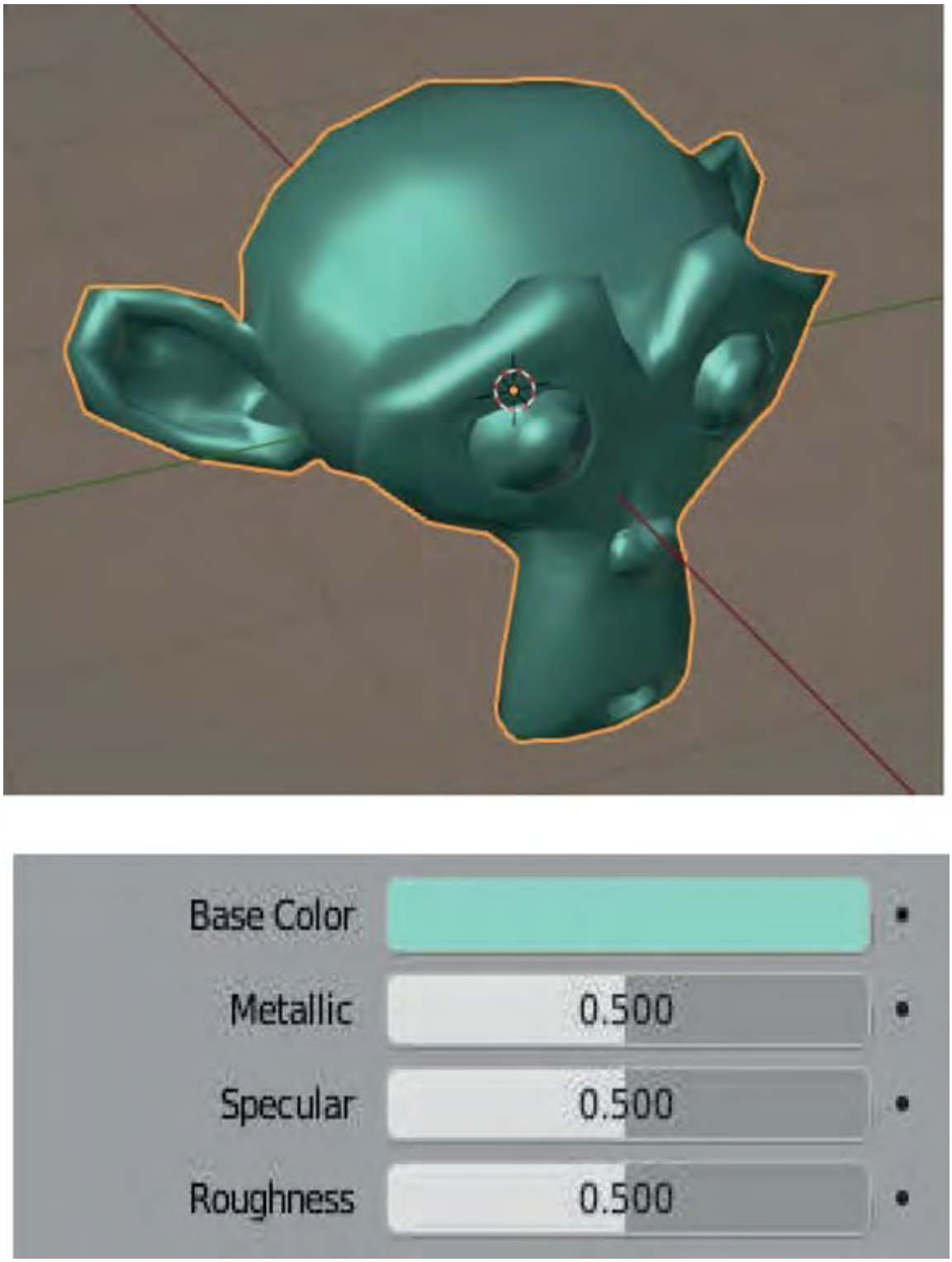


Figure 16.25

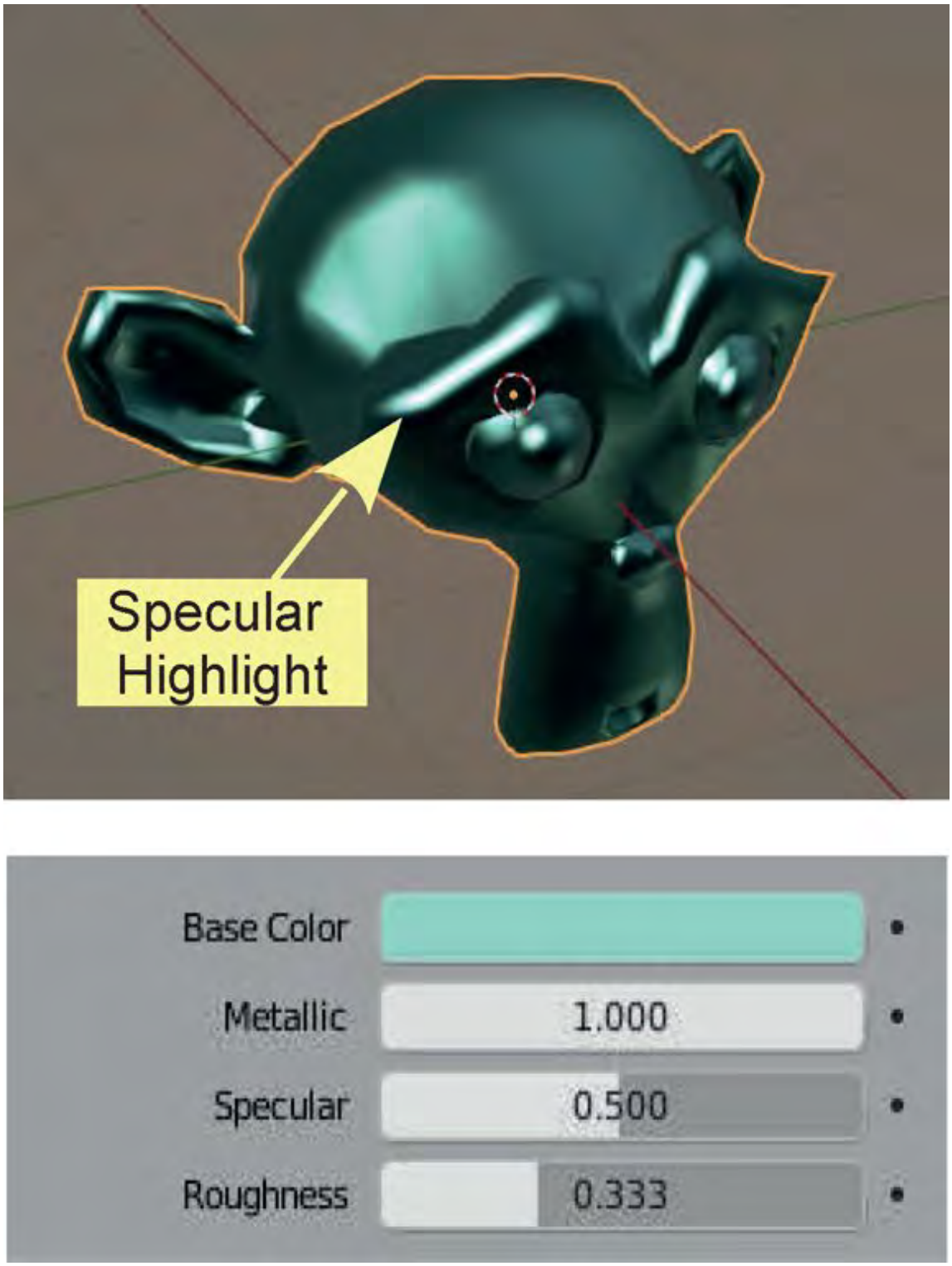

Specular
Highlight
Base Color
Metallic
1.000
Specular
0.500
Roughness
0.333

Figure 16.26

In the above, the **Specular** value has been maintained at 0.500 since the effect is very subtle. The Specular value is responsible for the highlights such as those seen on the tips of eyebrows and the pupils in the eyes.

16.10 Ligntiing Affects

How the Scene is illuminated (Chapter 15) and the **Viewport Shading** mode employed has an effect on how you see Materials in the 3D Viewport Editor. Consider the Monkey Object.

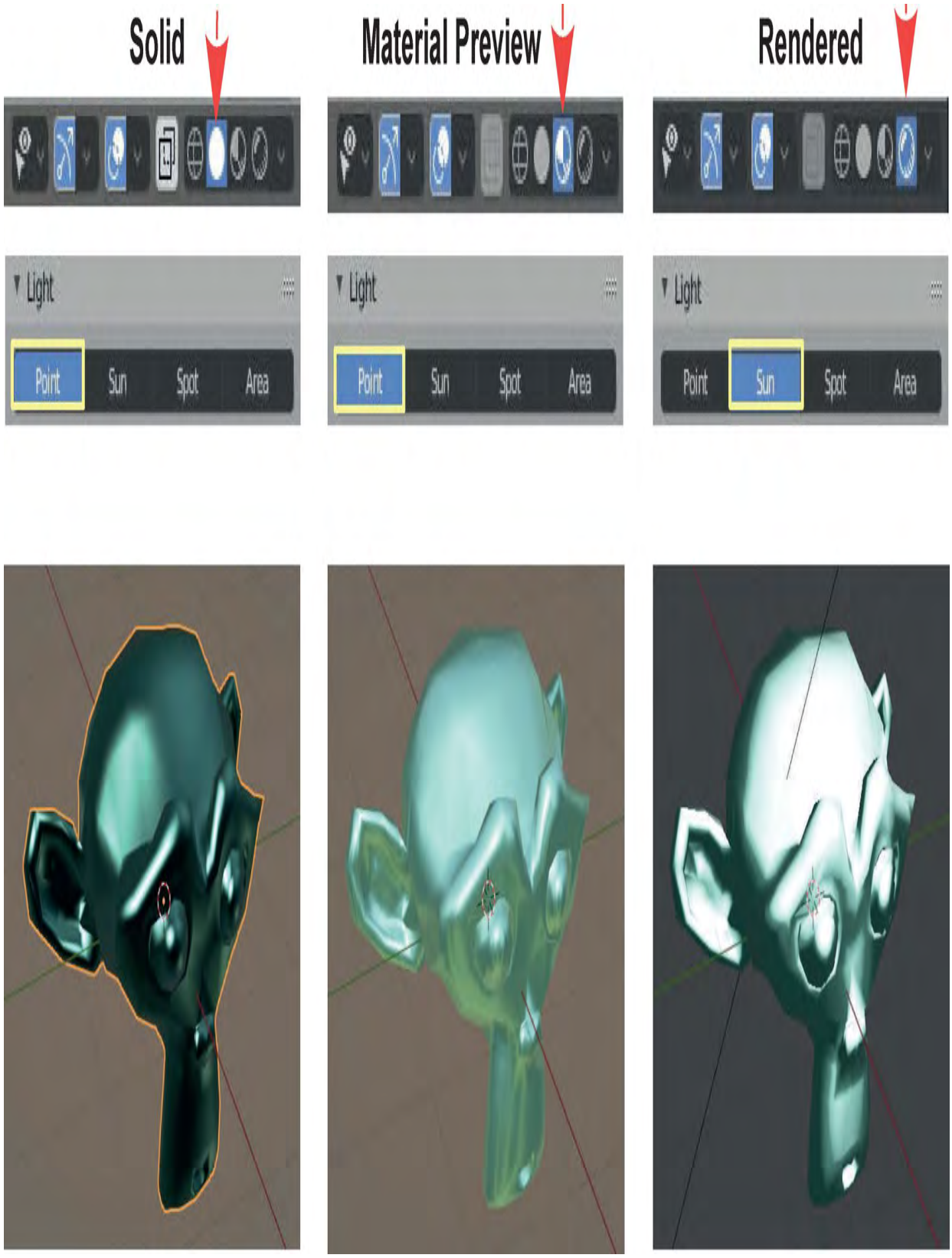

Solid
Material Preview
Rendered
Light
Point
Sun
Spot
Area
Light
Point
Sun
Spot
Area
Light
Point
Sun
Spot
Area

Figure 16.27

Material Properties (color) for all three images are the same (Figure 16.26).

In the preceding examples Materials have been demonstrated with the Material Node System **disabled.**

16.11 Materials using Nodes

When Blender first opens with the default Cube Object in the 3D Viewport Editor the Properties Editor, Material Properties show the controls for the Material of the Cube with the default Blender Material Node System **active.** This is indicated by the **Use Nodes** button being highlighted blue.

Figure 16.3 shows the default Node arrangement controlling the Material for the default Cube Object (the selected Object). This arrangement is displayed in the **Shader Editor.** Material Nodes are sometimes called Shaders. Figure 16.4 shows the Material Properties in the Properties Editor which for the default Material replicate the controls in the Principled BSDF Node. **Principled BSDF** is the name of the Node.

Blender has three different **Node Systems** which you see in the see in the Editor Type menu in the **3D Viewport Header**.

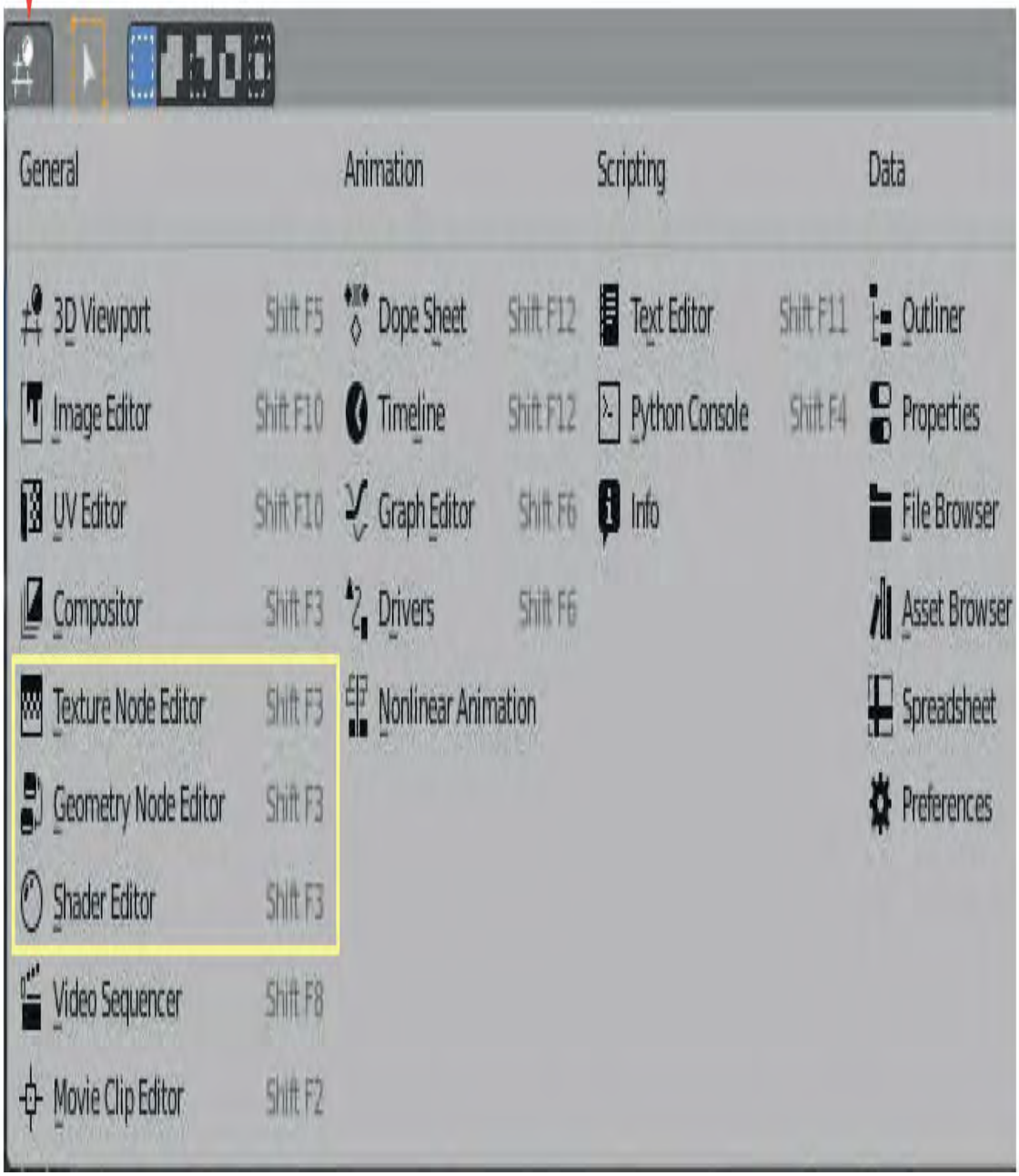


Figure 16.28

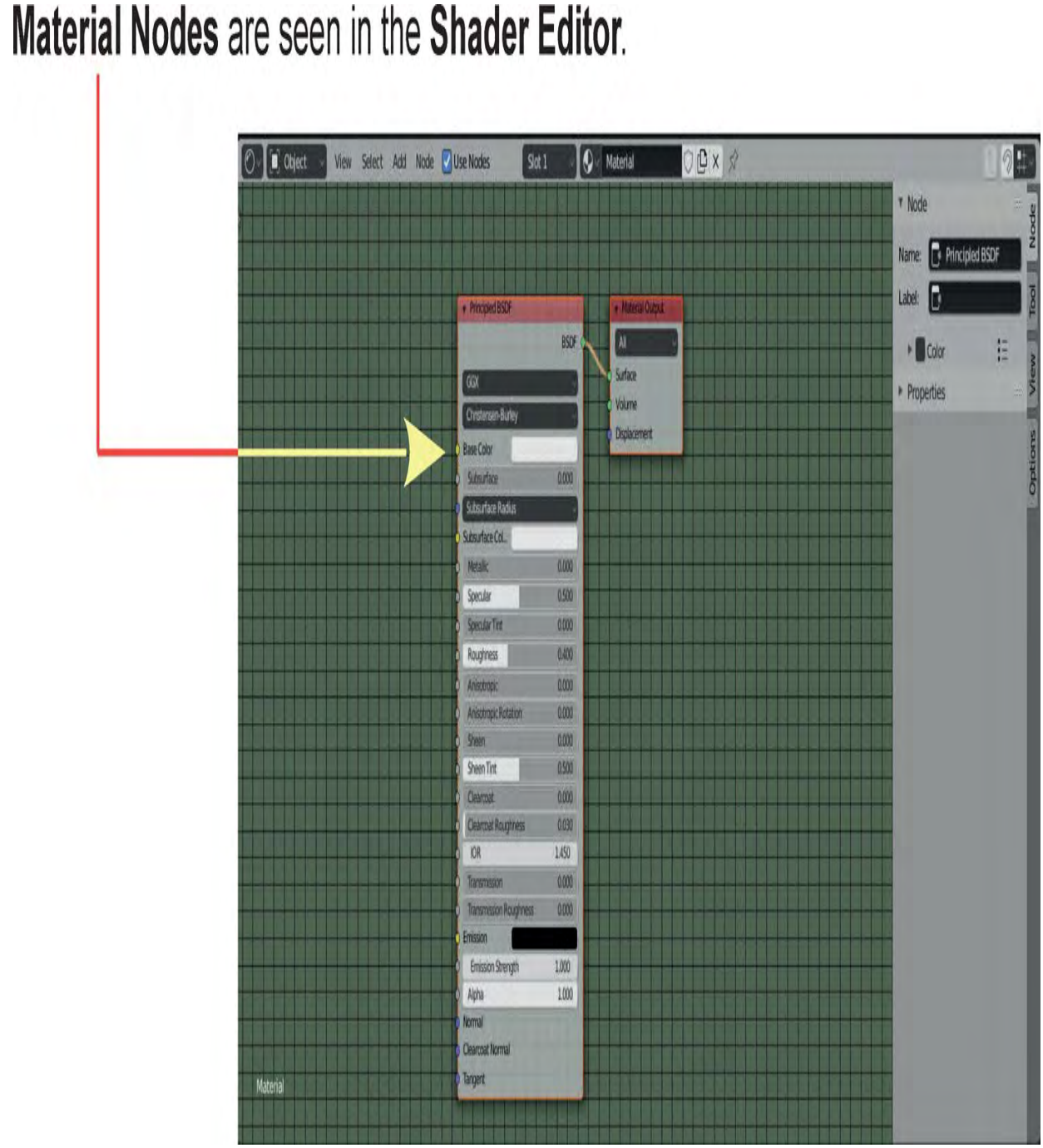


Figure 16.29

16.12 What Are Nodes?

Consider a **Node** as a point in a pipeline of information which contributes to a result. In the case of Material Nodes the result is the appearance of the surface of an Object in the 3D Viewport Editor. There are usually numerous Nodes

connected together producing the result, each of which may be disconnected, rearranged or replaced to vary the final display.

The **Node** is a graphical representation of computer data or instruction which is arranged in a pipeline. Think about mixing colors. The primary colors are Red, Green and Blue, which when mixed in equal proportions produce White (Figure 16.30).

In a **Node System** this would look like:

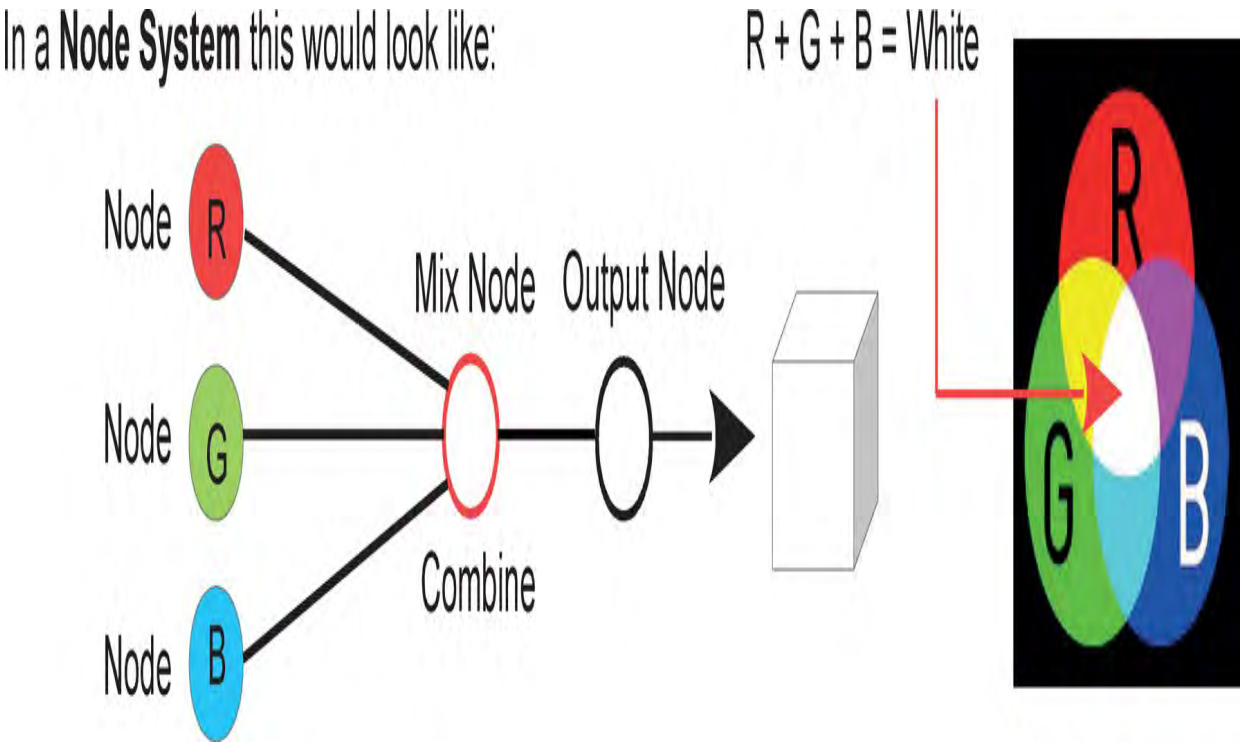


Figure 16.30

The Blender Node system looks like this:

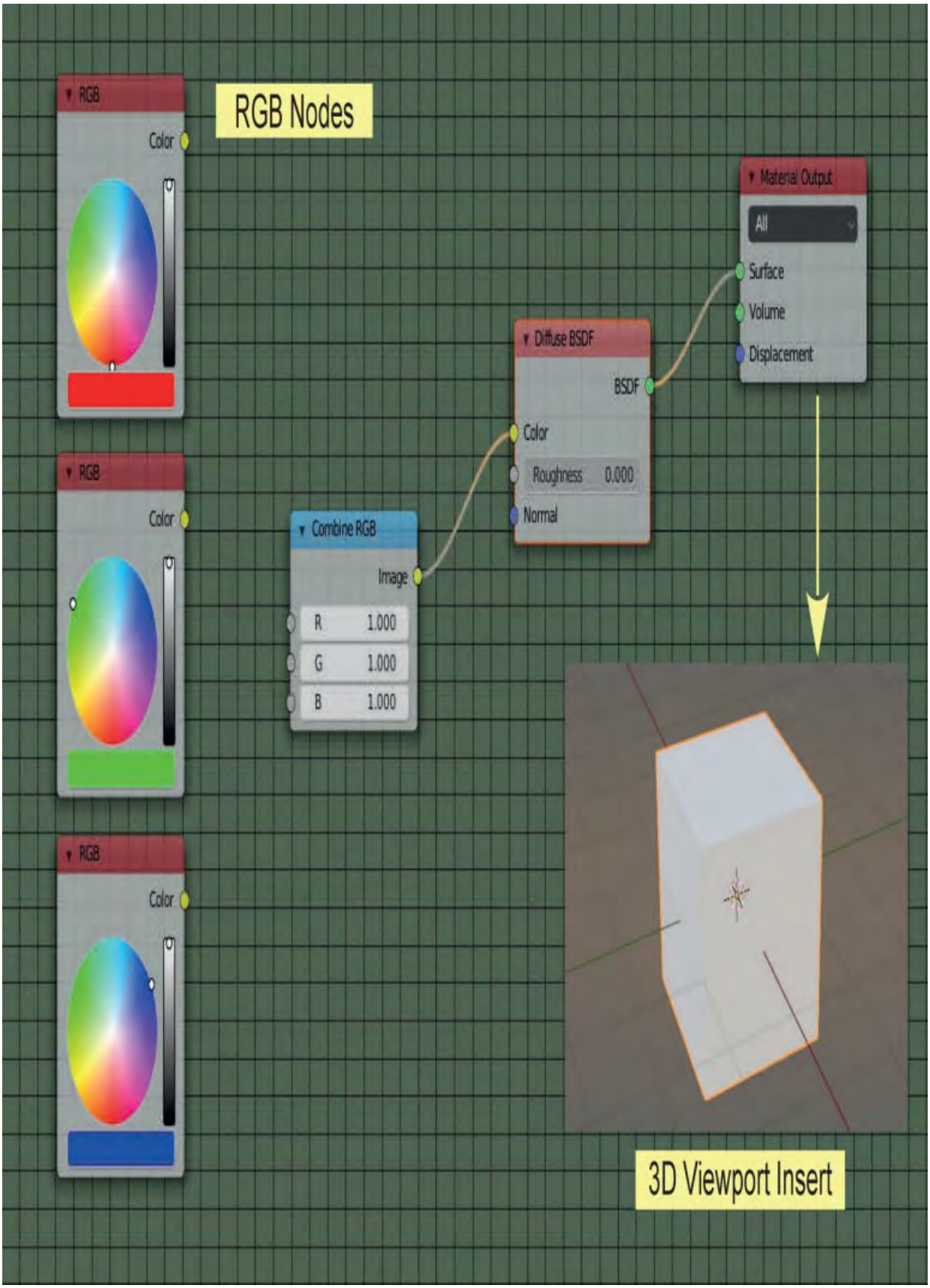

RGB Nodes
RGB
Color
RGB
Color
RGB
Color
Combine RGB
Image
R 1.000
G 1.000
B 1.000
Diffuse BSDF
BSDF
Color
Roughness 0.000
Normal
Material Output
All
Surface
Volume
Displacement
3D Viewport Insert

Figure 16.31

Note: Figure 16.31 is conceptional only (not strictly correct). The RGB Nodes are NOT connected and are represented instead by the **Combine RGB Node.** Red, Green and Blue values all 1.000.

A simple Node arrangement .

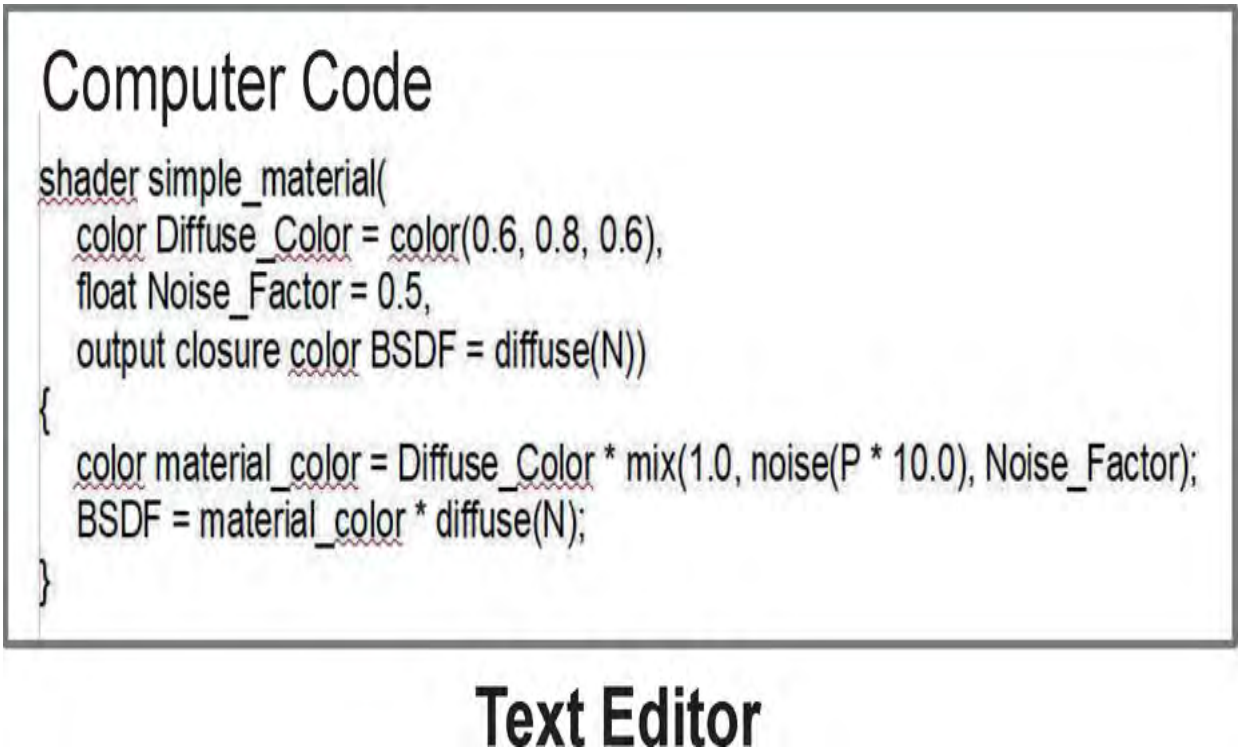


Figure 16.32

Computer Code written in **Python,** is represented by the **Diffuse BSDF Node,** which outputs data to the **Material Output Node.** This in turn applies Material (color) to the surface of the **Cube** Object in the 3D Viewport Editor.

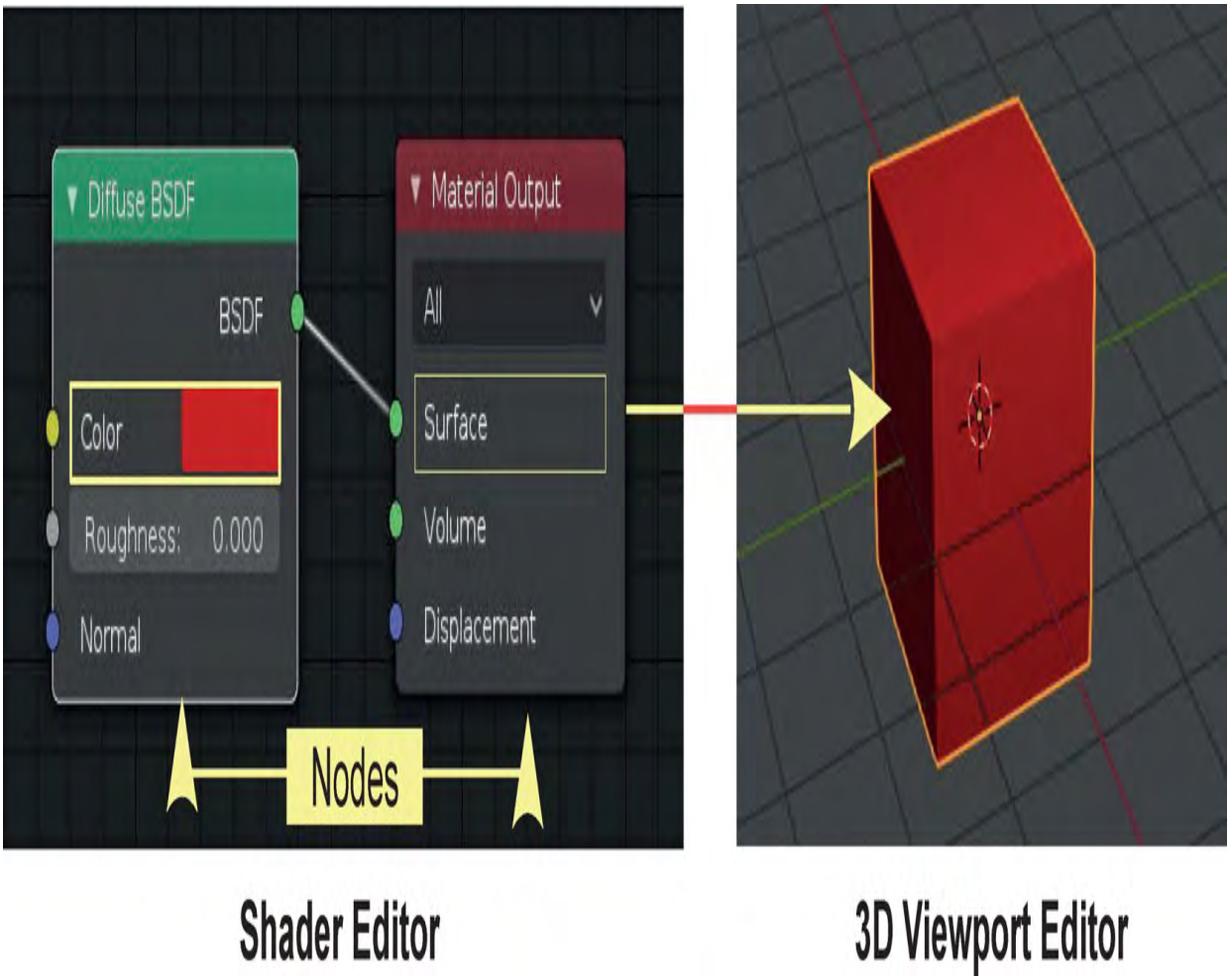


Figure 16.33

16.13 Accessing and Viewing Node Effects

To view Material Node effects it is advantageous to have an Object with a nice smooth surface.

Start over with a new Blender file. Delete the default Cube and replace it with a UV Sphere. Scale the UV Sphere down in the Z Axis forming a flat disk and set Shading Smooth (3D Viewport Editor Header in Object Mode — Click Object — Click Shade Smooth in the menu).

Note: How Figures and Diagrams display depends on how your version of Blender is configured (Reference Chapter 2 — 2.17).

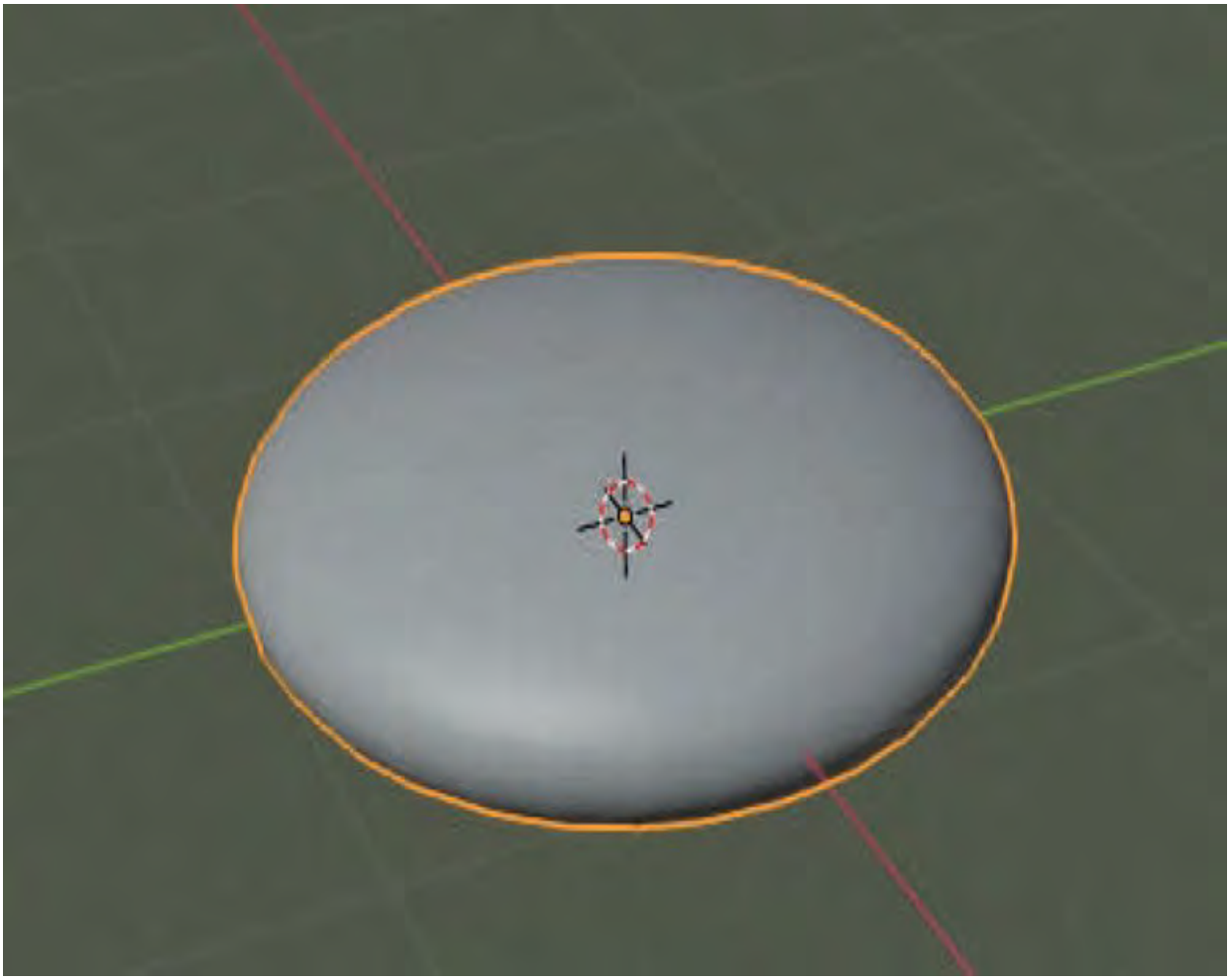


Figure 16.34

It is also advantageous to have the Scene adequately illuminated. Set the Scene lighting similar to that described in Chapter 15 - 15.7. You may also position a ground Plane below the modified Sphere.

Nodes are accessed in the **Shader Editor.** You could change the 3D Viewport Editor to the Shader Editor but it is useful to have both Editors displayed at the same time allowing you to see the effect in the 3D Viewport Editor as you compile a Node Arrangement.

To work with Nodes, divide the 3D Viewport Editor in two, horizontally and make the lower half the **Shader Editor** (Figure 16.35). Having both

Editors displayed allows you to see changes to Objects in the Scene as adjustments are made via the Nodes.

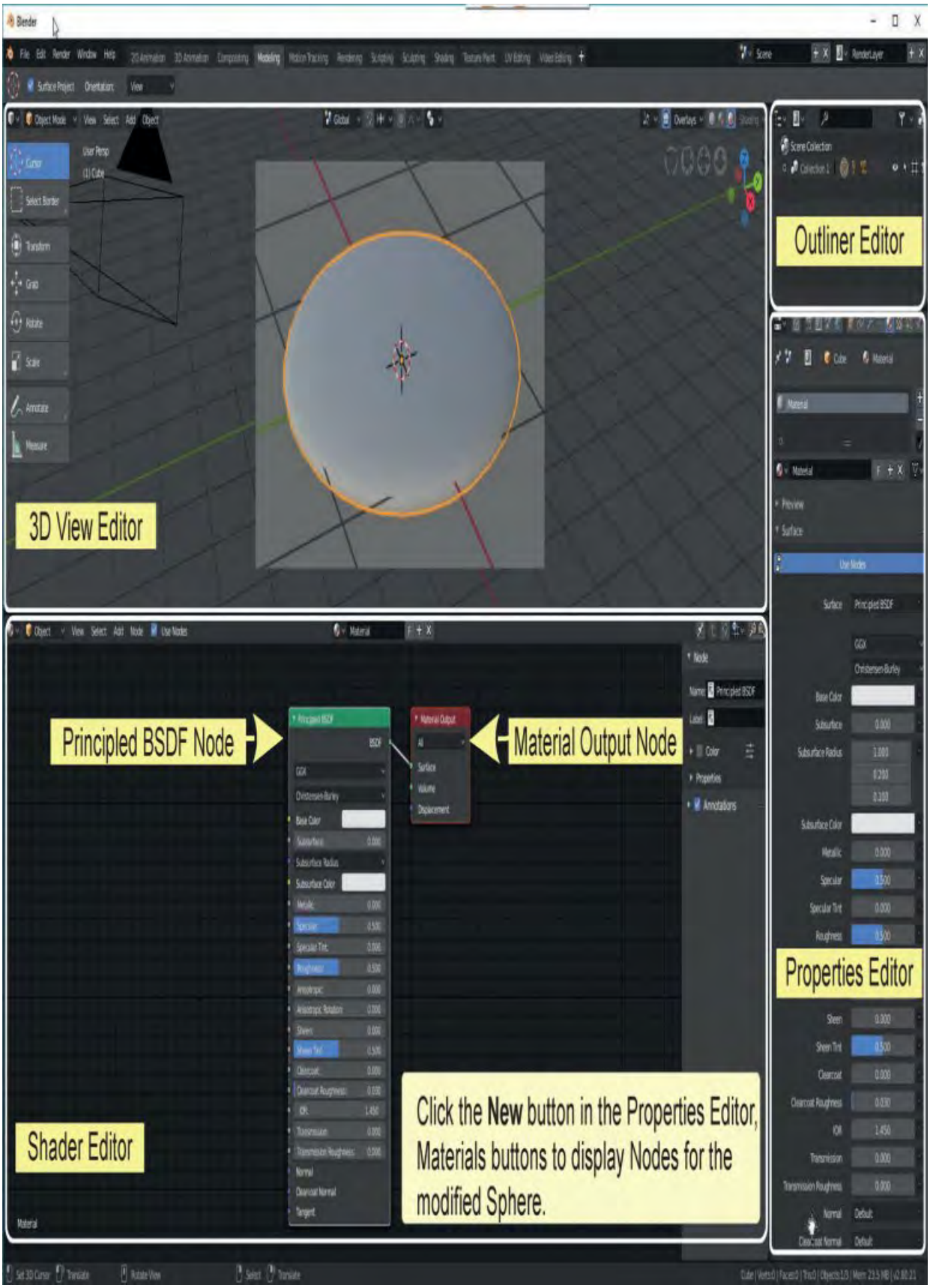

3D View Editor
Outliner Editor
Principled BSDF Node
Material Output Node
Shader Editor
Click the New button in the Properties Editor, Materials buttons to display Nodes for the modified Sphere.
Properties Editor

Figure 16.35

In the new Scene with the flat disk selected (New Object) go to the Material Properties in the Properties Editor and click **New.** Before clicking **New** the Shader Editor will be empty.

Note: If you open the Shader Editor with the default Cube Object selected in the 3D Viewport Editor, Nodes are displayed which control the Material color for the Cube. This occurs since the default Cube has a Material pre-applied (default Gray).

The Material Properties display with **Use Nodes** highlighted blue indicating that the Node System is active. In the **Shader Editor** two rectangles are displayed, one labelled **Principled BSDF** and the other **Material Output.** The rectangles are the Nodes. (zoom in — scroll MMB or press plus + or minus — on the Keyboard. You may also click MMB, hold and drag the Mouse to pan the view.)

Note: The **Principled BSDF Node** is not typical of all Nodes. This Node is what you might call a Super Node when you compare it with the **Diffuse BSDF Node** shown in Figure 16.33. It is shown here since it is displayed by default when Nodes are activated by clicking **Use Nodes** in the **Properties Editor.**

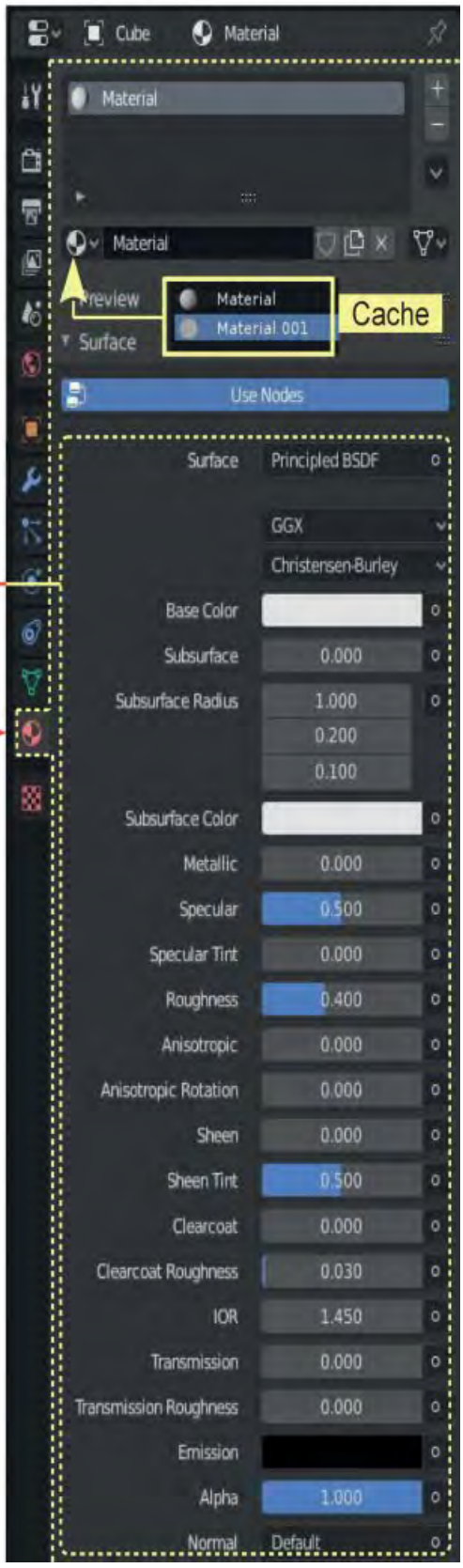
Note: The Principled BSDF Node displays values and controls identical to the controls in the Properties Editor, Material Properties when the Node System is active (Figure 16.35).

Compare the Principled BSDF Node to the content of the Properties Editor, Material Properties. You will see that the controls are identical.

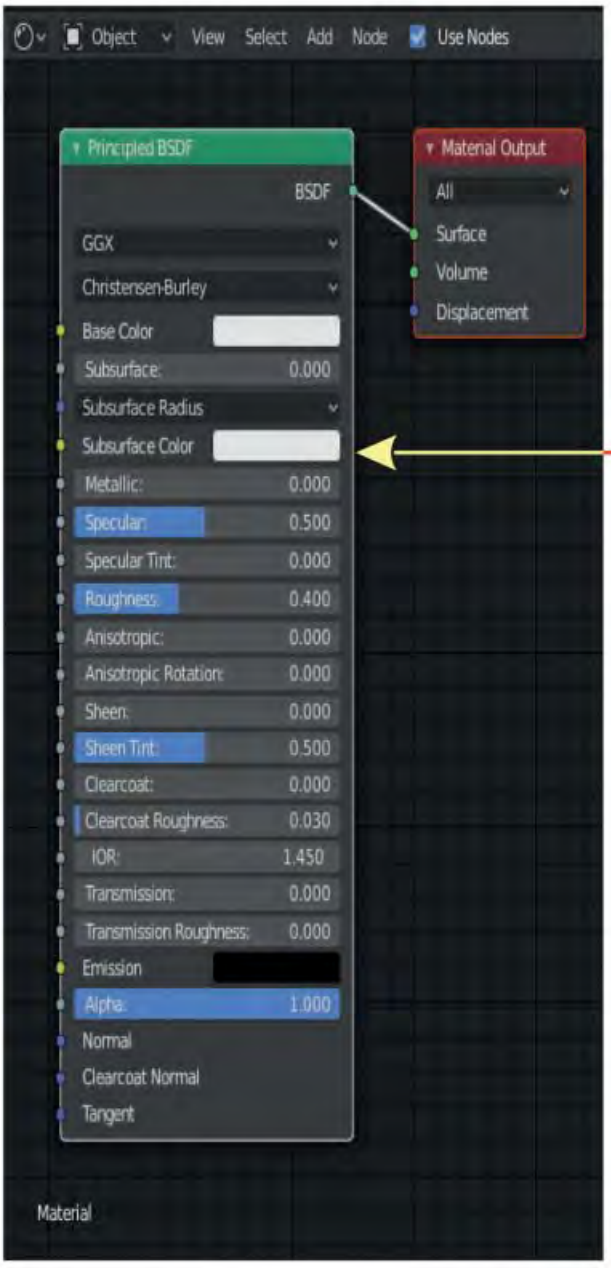
Properties Editor Material Properties

Nodes in the Shader Editor

Properties Editor Material Properties



Nodes in the Shader Editor



Note: The Material Properties Button is in the vertical column at the side of the Properties Editor.

Figure 16.36

To simplify the introduction to Nodes replace the Principled BSDF Node with the **Diffuse BSDF Node.** With the Mouse Cursor in the **Shader Editor,** press Alt + A Key on the Keyboard to deselect both Nodes. LMB click on a blank part of the Principled BSDF Node. The outline of the Node will display red. Press the **X Key** on the Keyboard to delete. Only the Material Output Node remains.

In the **Shader Editor Header** click **Add**, **Diffuse BSDF**.

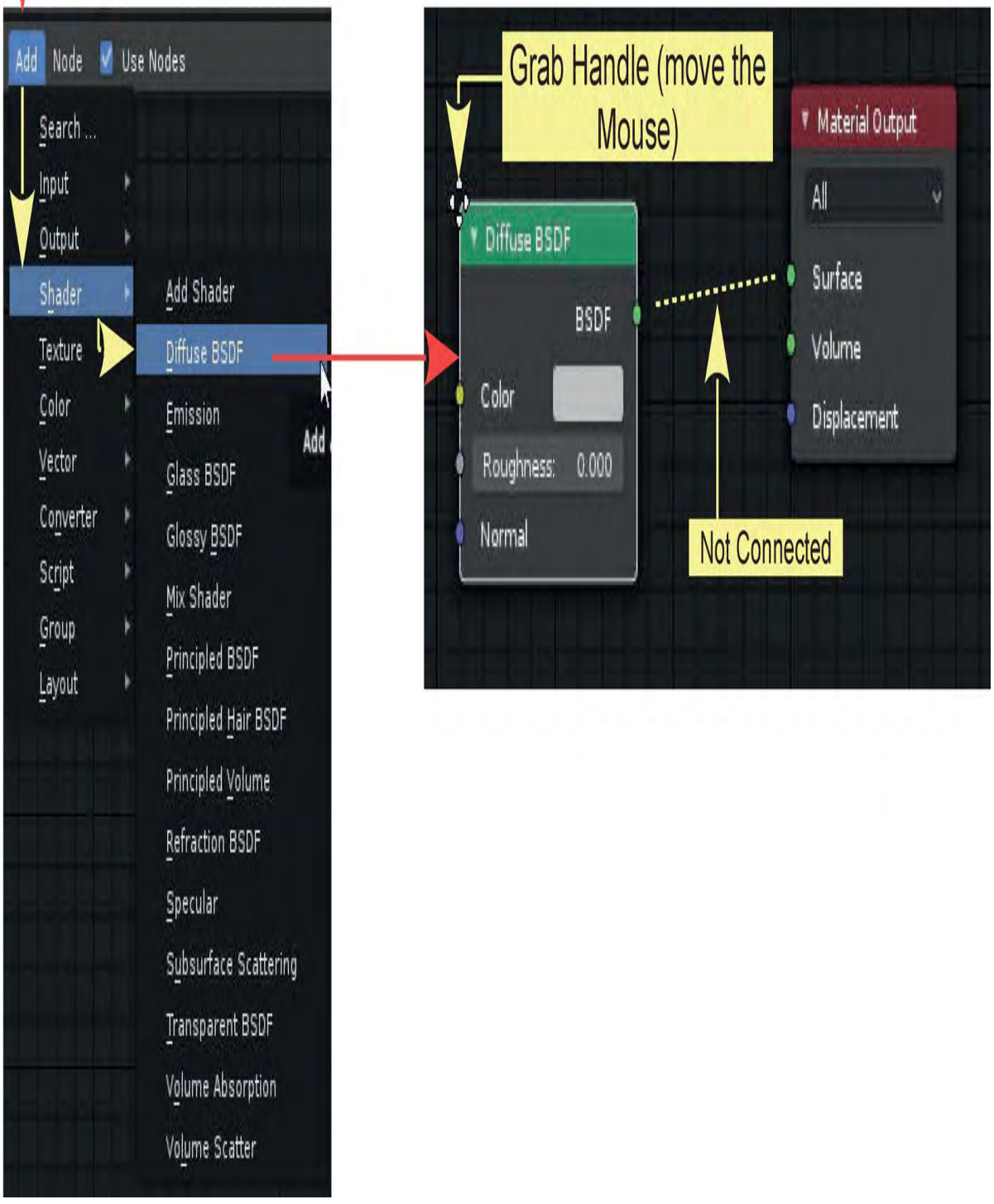
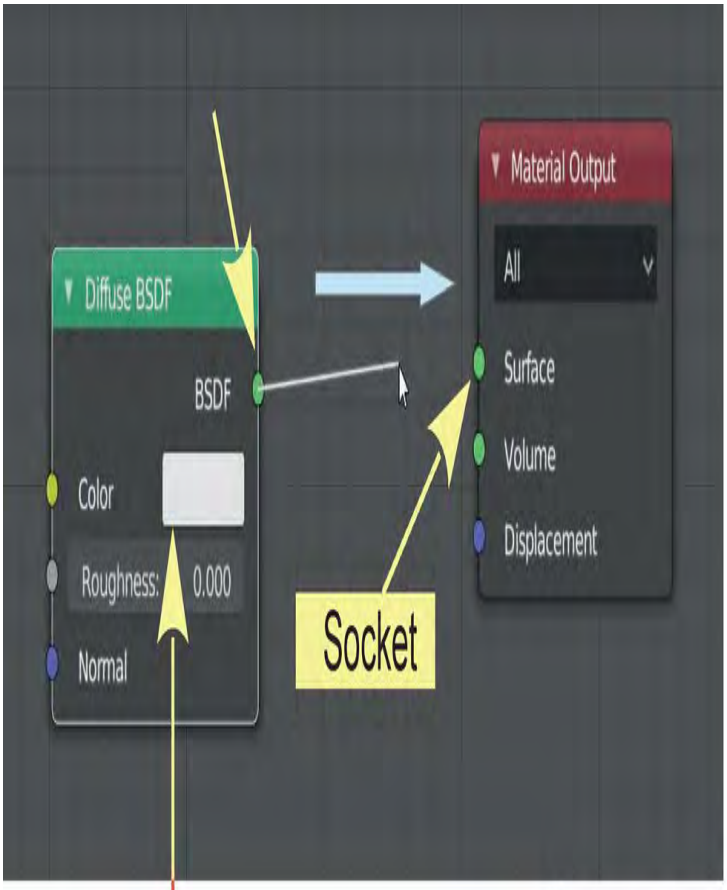


Figure 16.37

Click LMB on the Diffuse BSDF Node, hold and drag to displays a **Grab Handle** attached which follows Mouse movement allowing you to position the Node in the Editor. Release LMB to set in position. The handle is cancelled.

To adjust the position LMB click and hold on the Node (Grab Handle reappears) hold and drag the Mouse. Release.

The purpose of the **Diffuse BSDF Node** is to add a **Diffuse Material** (color) to the surface of the Object selected in the 3D Viewport Editor. To achieve this it has to be connected to the **Material Output Node.** At this point the Diffuse BSDF Node is not connected (Figure 16.37). To connect, click LMB on the green dot at the RHS of the Node, hold the Mouse button down and drag over to the green dot next to Surface at the LHS of the Material Output Node. Release the Mouse button to connect the Nodes (Figure 16.38 - 39). The connecting line is called a **Noodle.** The green dots are called **Sockets.**



Note: Clicking the Color bar in the Diffuse BSDF Node displays the color picker allowing you to change the surface color of the selected Object.

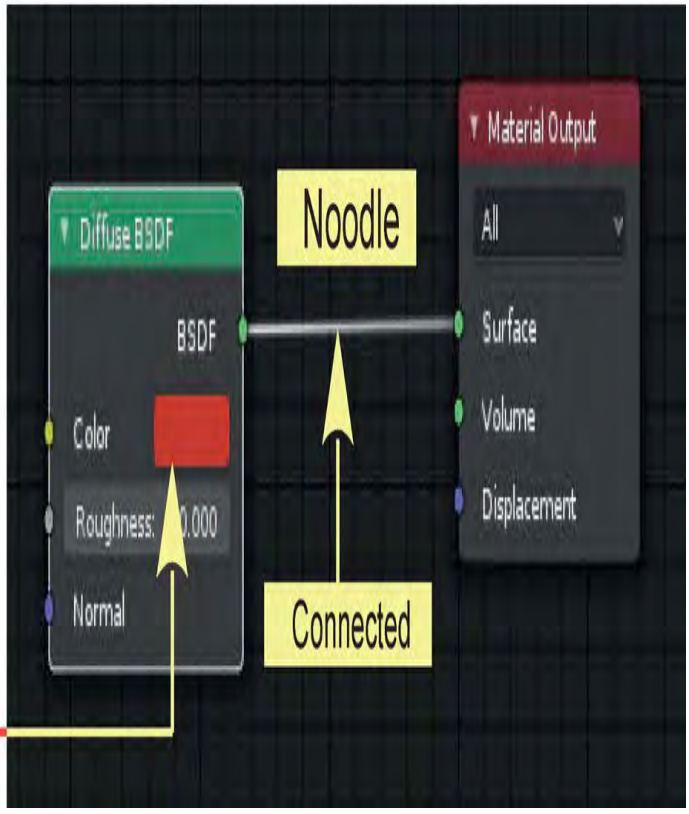


Figure 16.38
Figure 16.39

Diffuse: The color displays on the surface is the same, no matter where the light source or the camera is positioned in the Scene.

16.14 Noodle Curving

Note: The **Shader Editor** is sometimes called the **Node Editor.**

Figure 16.40

When **Noodles** are drawn they have a curve leading out and into the Sockets (Value: 1 - 10). This is purely a matter of preference and how much

curve is up to you.

16.15 The Shader Editor

Examine the content of the **Shader Editor Header.** The buttons generally display selection menus which, by and large, are self explanatory. One button of note is the **View button** with the **Toggle Sidebar (N)** and **Toggle Tool Shelf (T)** entries in the menu (Toggle display On / Off).

The **Sidebar** is the panel at the RHS of the Editor with Tabs for displaying information about the the selected Node, the selected Tool, the View and Options.

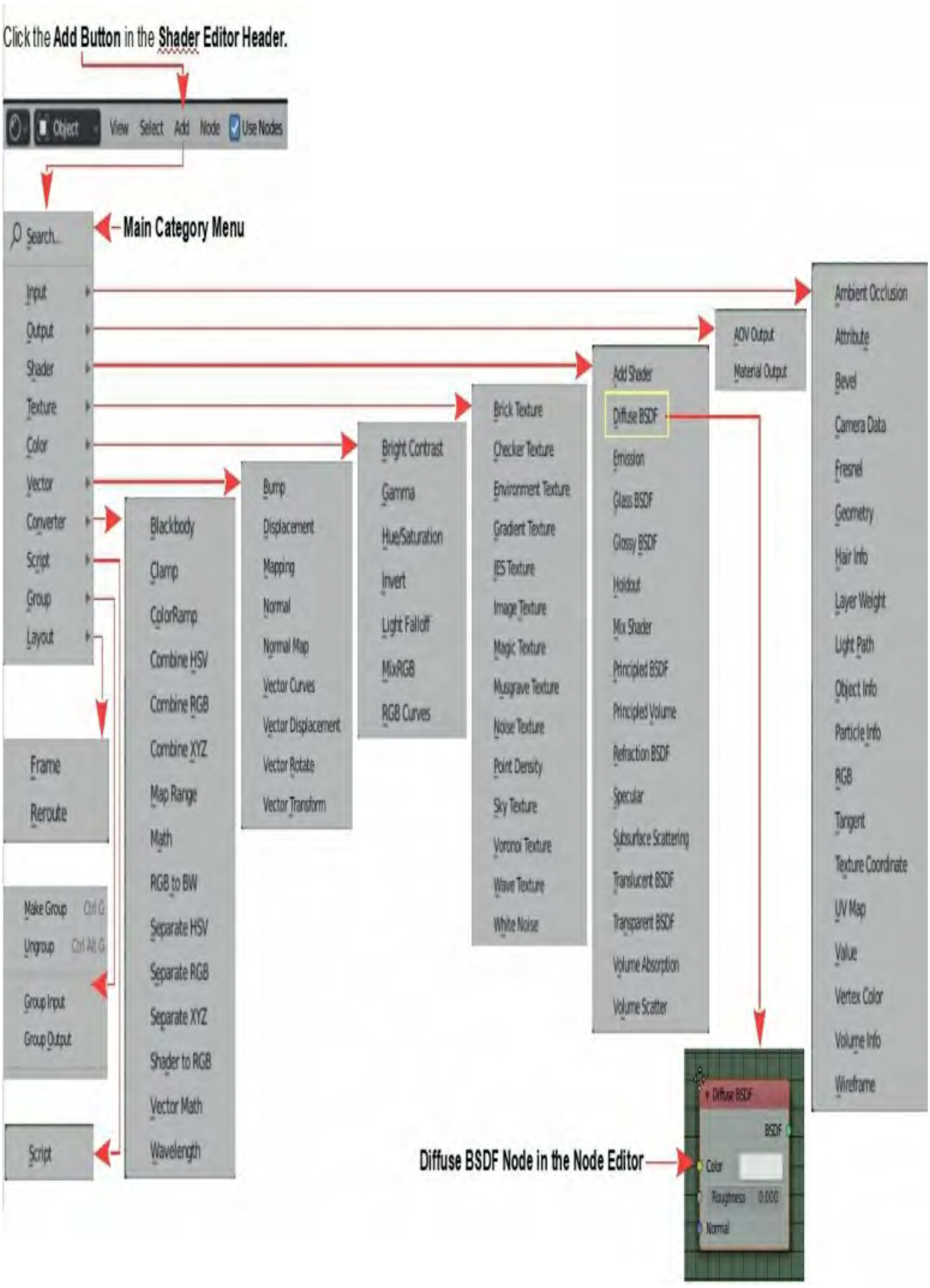
The Tool Shelf displays (press T Key) at the LHS of the Editor and contains three Tools; click to select (highlight blue).

Figure 16.41

The Add Buttor shows the **Shader Node Menu** 16.16 over.

16.16 The Shader Node Menu

Clicking the **Add Button** displays the **Main Category Menu**. Mouse overon a category to display a Node Selection Menu for the Category Click on a Node name to open the Node in the **Node Editor**, i.e click on Diffuse BSDF to enterthe **Diffuse BSDF** Node in the Node Editor


Click the Add Button in the Shader Editor Header.
Object
View
Select
Add
Node
Use Nodes
Search...
Main Category Menu
Input
Output
Shader
Texture
Color
Vector
Converter
Script
Group
Layout
Frame
Reroute
Make Group Ctrl G
Ungroup Ctrl Alt G
Group Input
Group Output
Script
Blackbody
Clamp
ColorRamp
Combine HSV
Combine RGB
Combine XYZ
Map Range
Math
RGB to BW
Separate HSV
Separate RGB
Separate XYZ
Shader to RGB
Vector Math
Wavelength
Bump
Displacement
Mapping
Normal
Normal Map
Vector Curves
Vector Displacement
Vector Rotate
Vector Transform
Bright Contrast
Gamma
Hue/Saturation
Invert
Light Falloff
MixRGB
RGB Curves
Brick Texture
Checker Texture
Environment Texture
Gradient Texture
IES Texture
Image Texture
Magic Texture
Musgrave Texture
Noise Texture
Point Density
Sky Texture
Voronoi Texture
Wave Texture
White Noise
Add Shader
Diffuse BSDF
Emission
Glass BSDF
Glossy BSDF
Holdout
Mix Shader
Principled BSDF
Principled Volume
Refraction BSDF
Specular
Subsurface Scattering
Translucent BSDF
Transparent BSDF
Volume Absorption
Volume Scatter
AOV Output
Material Output
Ambient Occlusion
Attribute
Bevel
Camera Data
Fresnel
Geometry
Hair Info
Layer Weight
Light Path
Object Info
Particle Info
RGB
Tangent
Texture Coordinate
UV Map
Value
Vertex Color
Volume Info
Wireframe
Diffuse BSDF
BSDF
Color
Roughness 0.000
Normal
Diffuse BSDF Node in the Node Editor

Having the selection menu displayed will assist when following tutorials and replicating Node Arrangements.

16.17 Adding Nodes

Figure 16.37 introduced the procedure for adding Nodes into the Editor by entering a Diffuse BSDF Node. Clicking the **Add button** in the Shader Editor Header opens a category list where you select a category to display the relevant Nodes. The categories assist when you are conversant with the function of each Node and how to arrange them to produce an effect. Knowing which Node to select and how to connect comes from experience, which in the beginning, is gained by following tutorials and copying and experimenting. In practical terms the Add button in the Shader Editor Header is the gateway to the maze of Nodes. When following tutorials you find a particular Node by either entering the name of the Node in the search panel at the top of the Category list or by navigating through the different categories.

Figure 16.41 on the preceding page displays the different categories which will assist in your search.

Becoming proficient in the use of Nodes comes with a certain amount of experimentation and organisation. When you have created a Node arrangement which produces a result, save it in a Blender file for future use. You may wish to use it again in another project or use it as a starting point for further development.

16.18 The Shading Workspace

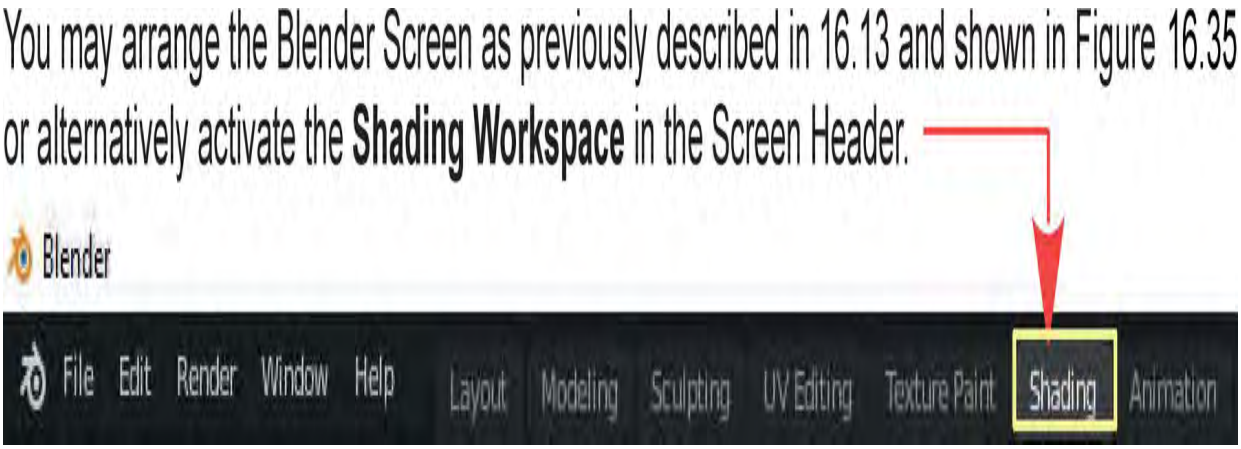

You may arrange the Blender Screen as previously described in 16.13 and shown in Figure 16.35 or alternatively activate the **Shading Workspace** in the Screen Header.
Blender
File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation

Figure 16.42

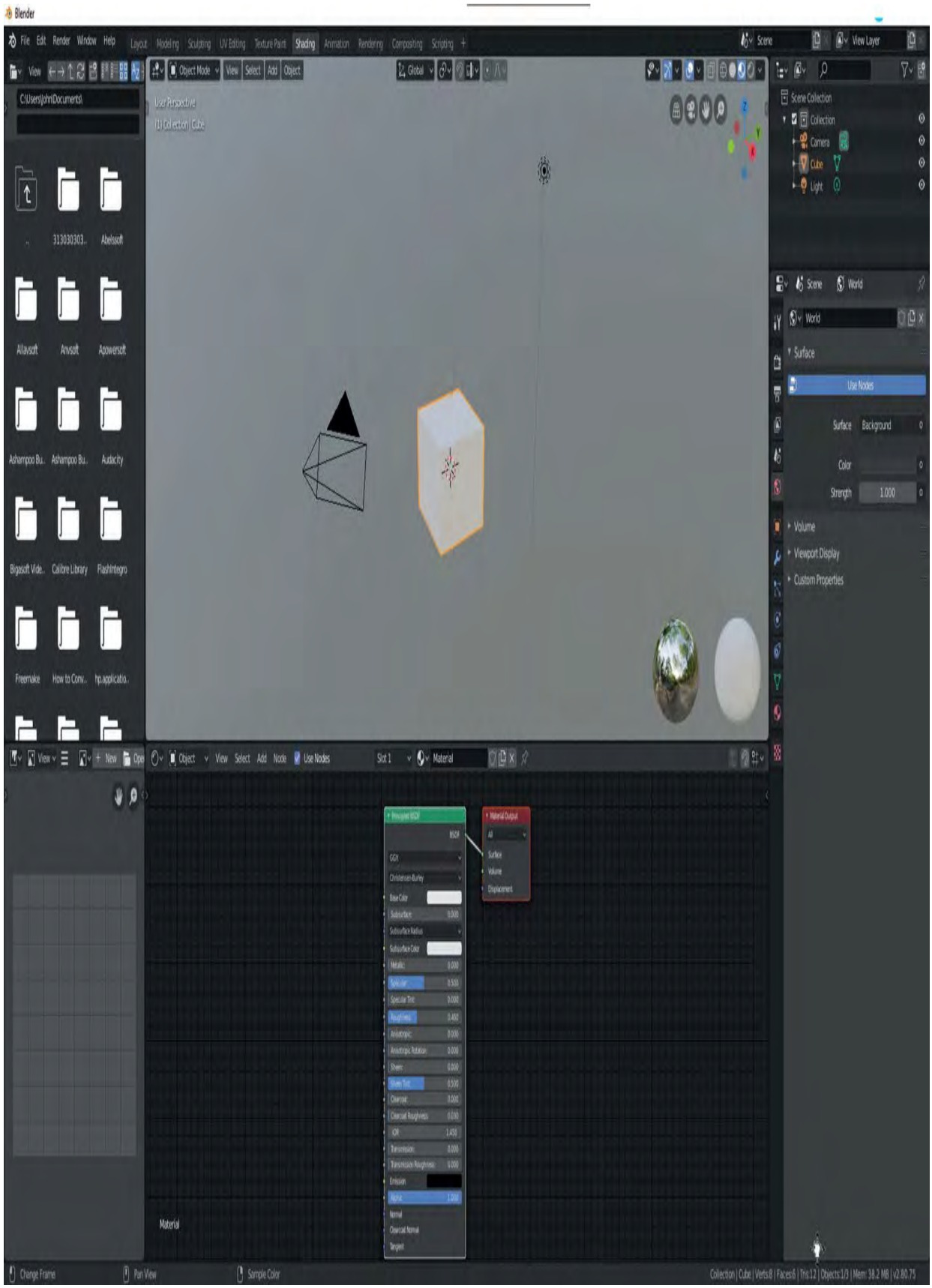

Blender
File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Scripting
Scene
View Layer
Object Mode View Select Add Object
Global
User Perspective
(1) Collection | Cube
Scene Collection
Collection
Camera
Cube
Light
Scene World
World
Surface
Use Nodes
Surface Background
Color
Strength 1.000
Volume
Viewport Display
Custom Properties
Object View Select Add Node Use Nodes
Slot 1
Material
Material Output
All
Surface
Volume
Displacement
Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface
Subsurface Radius
Subsurface Color
Metallic
Specular
Specular Tint
Roughness
Anisotropic
Anisotropic Rotation
Sheen
Sheen Tint
Clearcoat
Clearcoat Roughness
IOR
Transmission
Transmission Roughness
Emission
Alpha
Normal
Clearcoat Normal
Tangent
Material
Change Frame
Pan View
Sample Color
Collection | Cube | Verts:8 | Faces:6 | Tris:12 | Objects:1/3 | Mem: 38.2 MB | v2.80.75

Figure 16.43

The **Shading Workspace** displays with six Editors forming a working environment for working with Nodes. At this stage it is suggested you work with the Screen Arrangement as shown in Figure 16.35.

16.19 Scene Arrangements

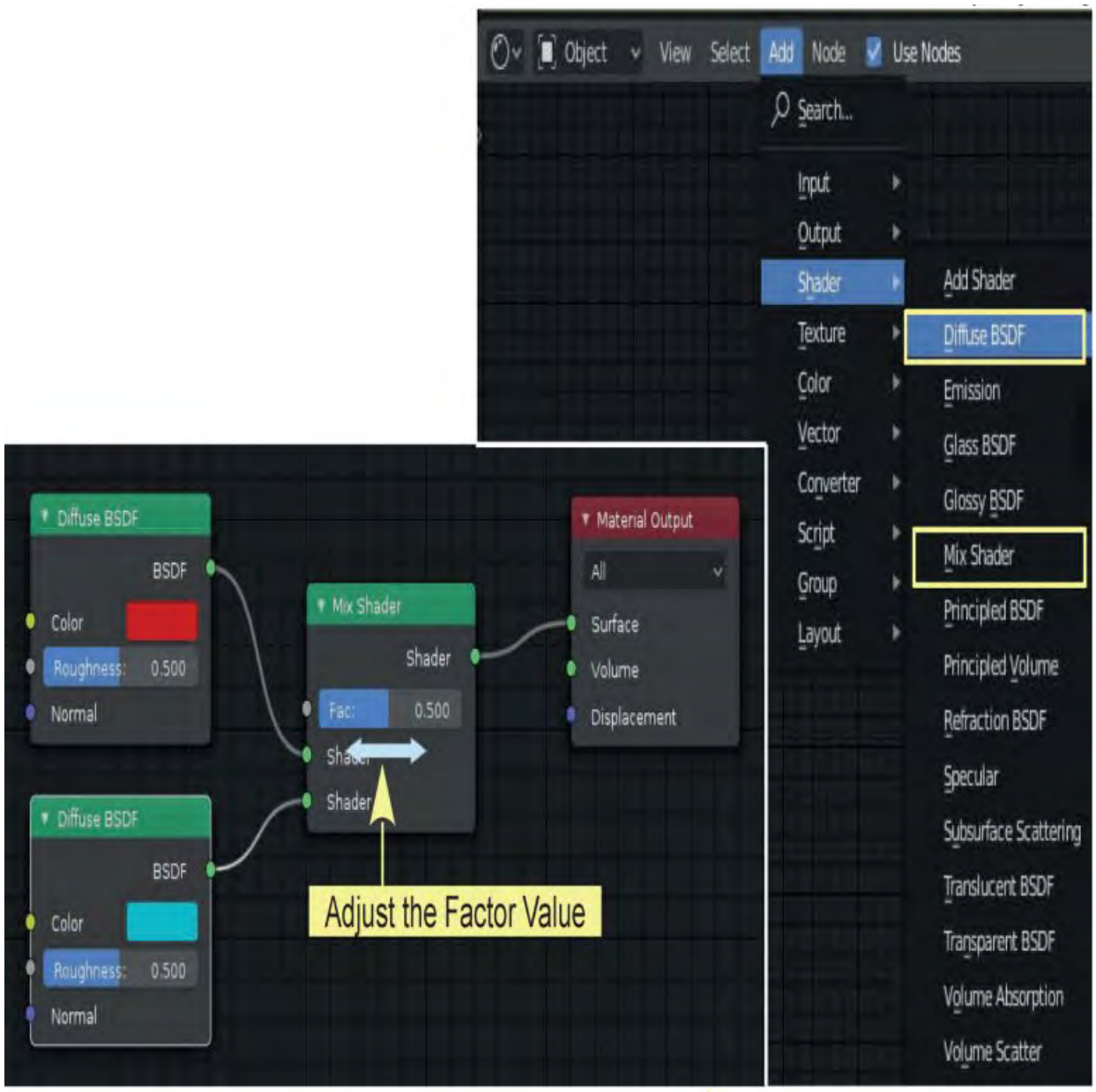
Before trying examples of Node Arrangements set up a Scene as previously described, with a smooth UV Sphere scaled down on the Z Axis forming a flat disc. Have the disc sitting above a Plane and place Lamps similar to those in Chapter 15 — 15.7 (Basic Lighting Arrangement). Have the 3D View Editor in Camera View with Material Preview Viewport Shading and Scene Lights checked in the Sub Options. Alternatively be in Rendered Viewport Shading Mode.

To understand how Nodes operate replicate the Node arrangements in the following examples.

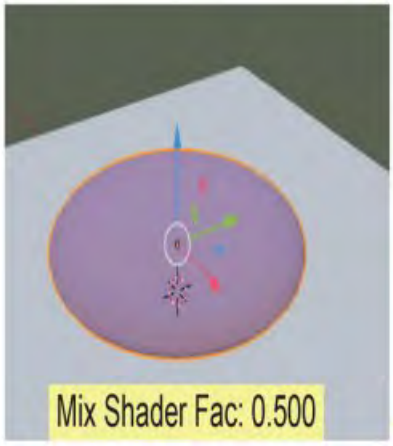
16.20 Mixing Material cxarinpia.

With the Node arrangement in Figure 16.44 two Material colors (red and blue) may be mixed together. The colors for mixing are selected in the **Diffuse BSDF** Nodes and mixed by adjusting the Factor slider in the **Mix Shader** Node.

In this instance the Nodes are found in the **Add menu** in the **Shader** category.


Object
View
Select
Add
Node
Use Nodes
Search...
Input
Output
Shader
Texture
Color
Vector
Converter
Script
Group
Layout
Add Shader
Diffuse BSDF
Emission
Glass BSDF
Glossy BSDF
Mix Shader
Principled BSDF
Principled Volume
Refraction BSDF
Specular
Subsurface Scattering
Translucent BSDF
Transparent BSDF
Volume Absorption
Volume Scatter
Diffuse BSDF
BSDF
Color
Roughness: 0.500
Normal
Mix Shader
Shader
Fac: 0.500
Shader
Shader
Material Output
All
Surface
Volume
Displacement
Diffuse BSDF
BSDF
Color
Roughness: 0.500
Normal
Adjust the Factor Value

Mix Shader Fac: 0.000


Mix Shader Fac: 0.500

Mix Shader Fac: 1.000

Figure 16.44

16.21 The Principled BSDF Node

The default Node arrangement in the Shader Editor for Materials shows the **Principled BSDF Node** connected to the **Material Output Node** when the New button is pressed in the Properties Editor, Material Properties. This applies a Material to the selected Object entered into the Scene.

Figures 16.36 - 16.45 show that the Principled BSDF Node replicates the controls displayed in the Properties Editor. Adjusting controls in the Properties Editor or in the Principled BSDF Node affects the selected Object in the 3D Viewport Editor i.e. selecting a Base Color and adjusting the Metallic, Specular and Roughness values.

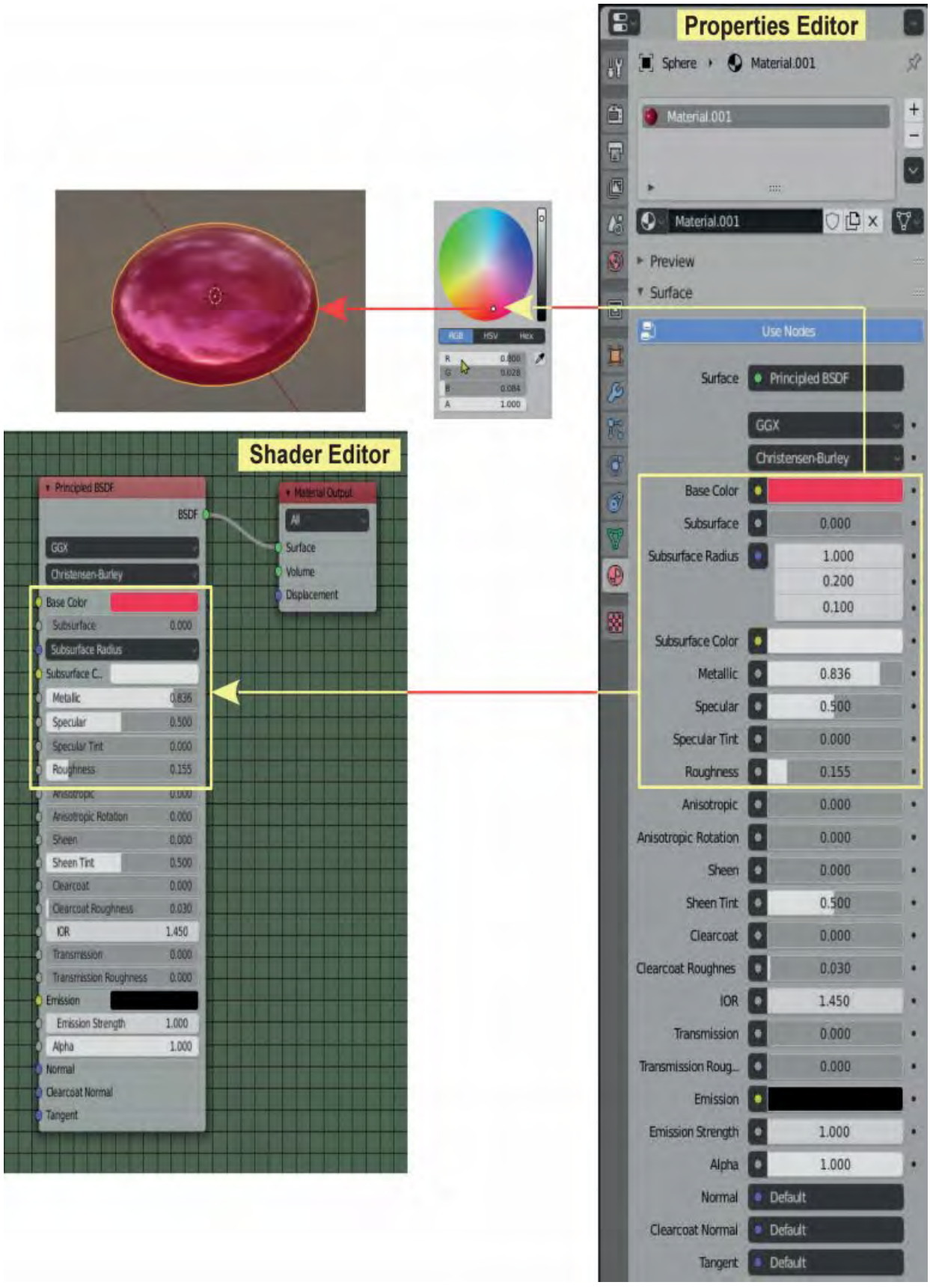

Properties Editor
Sphere
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
1.000
0.200
0.100
Subsurface Color
Metallic
0.836
Specular
0.500
Specular Tint
0.000
Roughness
0.155
Anisotropic
0.000
Anisotropic Rotation
0.000
Sheen
0.000
Sheen Tint
0.500
Clearcoat
0.000
Clearcoat Roughnes
0.030
IOR
1.450
Transmission
0.000
Transmission Roug...
0.000
Emission
Emission Strength
1.000
Alpha
1.000
Normal
Default
Clearcoat Normal
Default
Tangent
Default
RGB
HSV
Hex
R
0.800
G
0.028
B
0.084
A
1.000
Shader Editor
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
Subsurface C..
Metallic
0.836
Specular
0.500
Specular Tint
0.000
Roughness
0.155
Anisotropic
0.000
Anisotropic Rotation
0.000
Sheen
0.000
Sheen Tint
0.500
Clearcoat
0.000
Clearcoat Roughness
0.030
IOR
1.450
Transmission
0.000
Transmission Roughness
0.000
Emission
Emission Strength
1.000
Alpha
1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement

Figure 16.45

Remember: When using Nodes you must have the 3D Viewport Editor in Material Preview or Rendered Viewport Shading Mode to see results.

This default Node arrangement may be considered as a starting point for generating a Material. Additional Nodes are entered into the Shader Editor and connected to the input sockets at the left hand side of the Principled BSDF Node (Figure 16.46).

There are two methods of adding Nodes; click the Add button in the Shader Editor Header and select from the menus, then manually connect sockets or click the buttons at the left hand side of the values in the Properties Editor. In the later case the Node being added is automatically connected to the Principled BSDF Node.

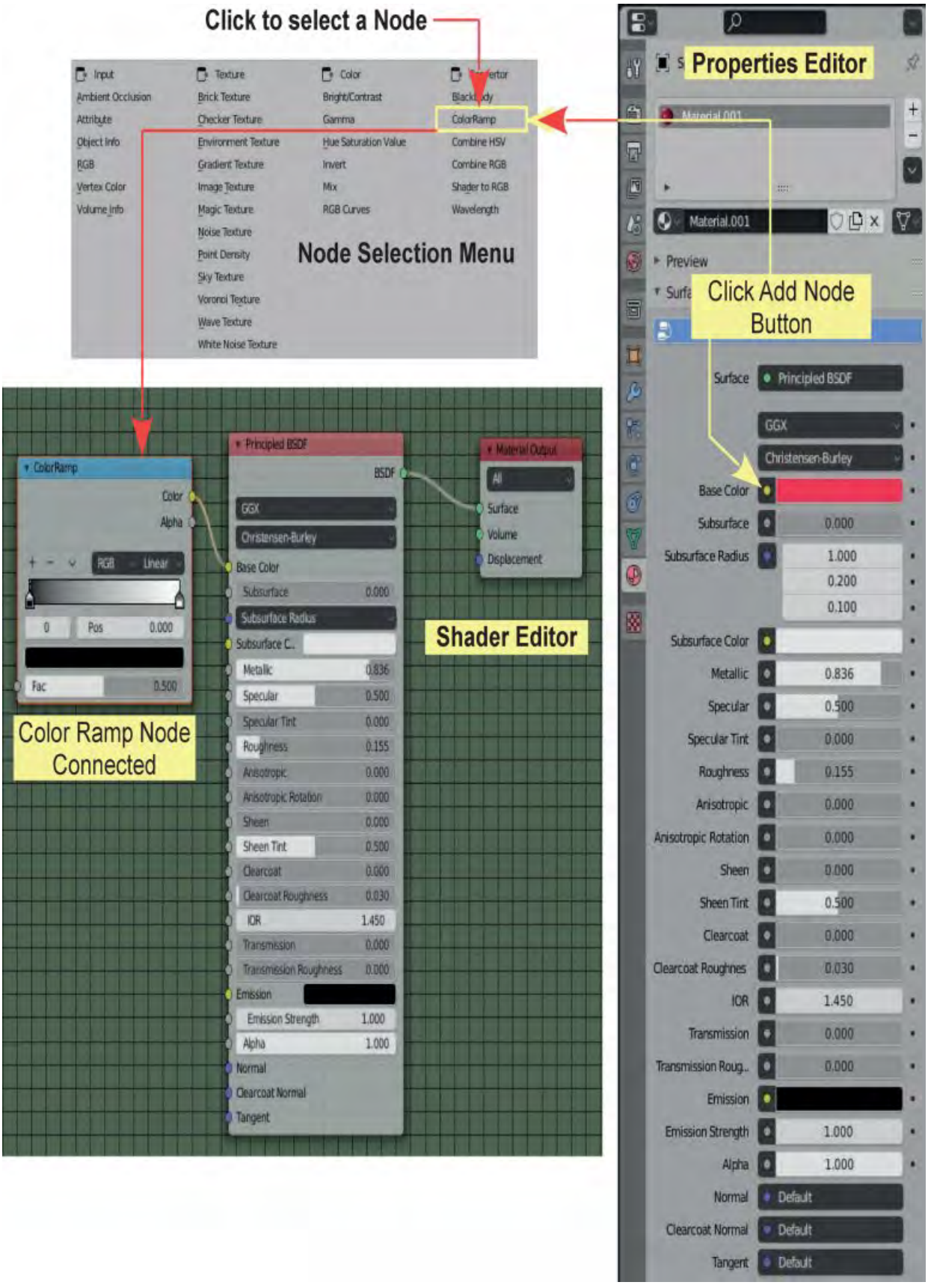

Click to select a Node
Input
Ambient Occlusion
Attribute
Object Info
RGB
Vertex Color
Volume Info
Texture
Brick Texture
Checker Texture
Environment Texture
Gradient Texture
Image Texture
Magic Texture
Noise Texture
Point Density
Sky Texture
Voronoi Texture
Wave Texture
White Noise Texture
Color
Bright/Contrast
Gamma
Hue Saturation Value
Invert
Mix
RGB Curves
ColorRamp
Combine HSV
Combine RGB
Shader to RGB
Wavelength
Node Selection Menu
Properties Editor
Material.001
Preview
Click Add Node Button
ColorRamp
Color
Alpha
RGB
Linear
Pos
0.000
Fac
0.500
Color Ramp Node Connected
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface
Subsurface Radius
Subsurface C..
Metallic
Specular
Specular Tint
Roughness
Anisotropic
Anisotropic Rotation
Sheen
Sheen Tint
Clearcoat
Clearcoat Roughness
IOR
Transmission
Transmission Roughness
Emission
Emission Strength
Alpha
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement
Shader Editor
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
1.000
0.200
0.100
Subsurface Color
Metallic
0.836
Specular
0.500
Specular Tint
0.000
Roughness
0.155
Anisotropic
0.000
Anisotropic Rotation
0.000
Sheen
0.000
Sheen Tint
0.500
Clearcoat
0.000
Clearcoat Roughnes
0.030
IOR
1.450
Transmission
0.000
Transmission Roug..
0.000
Emission
Emission Strength
1.000
Alpha
1.000
Normal
Default
Clearcoat Normal
Default
Tangent
Default

Figure 16.46

Note: The display in the Node Selection Menu depends on the function selected in the Properties Editor.

Figure 16.47 shows the **Principled BSDF Node** in the **Shader Editor** listing some of the functions contained in the Node.

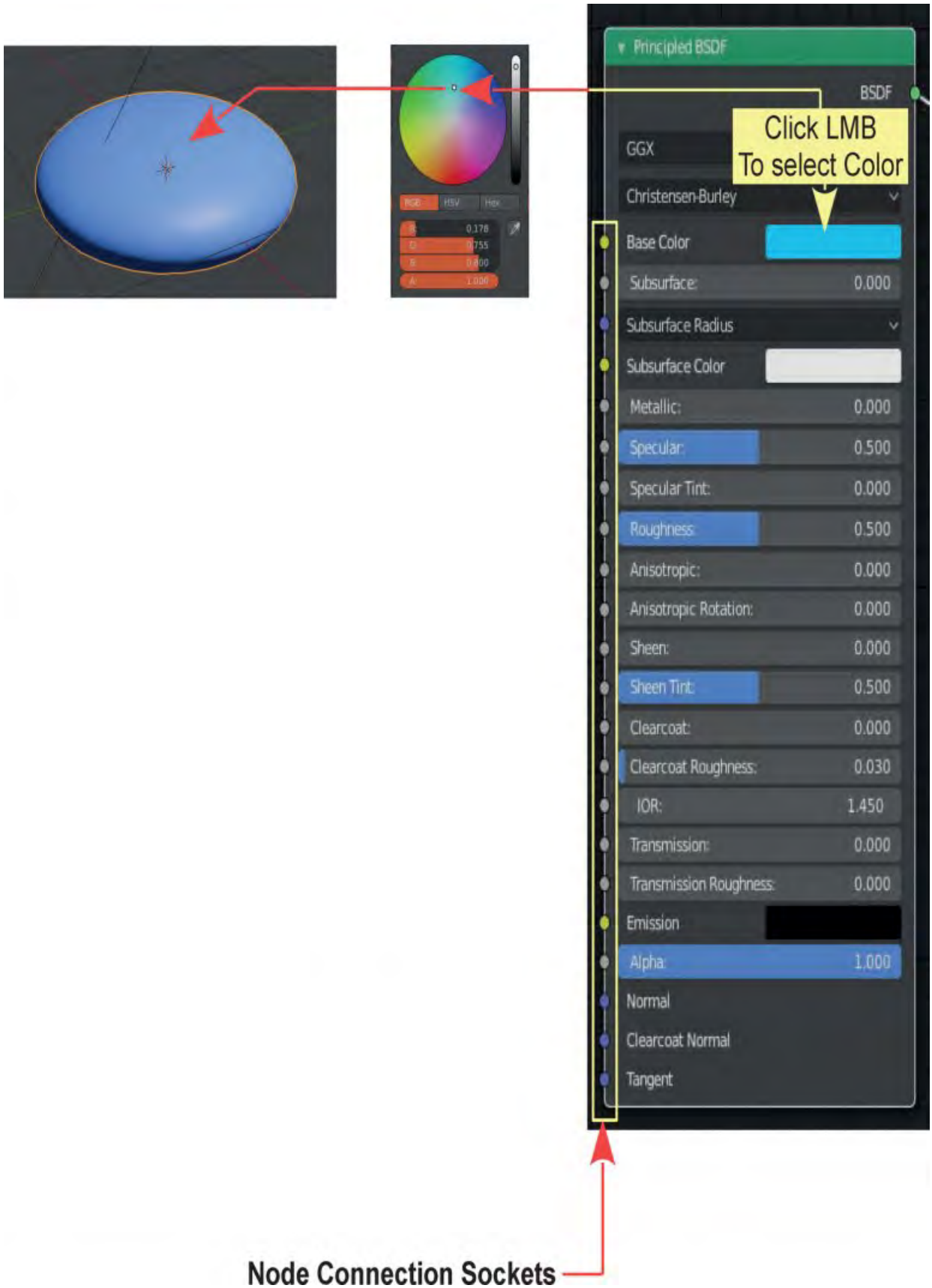

Principled BSDF
BSDF
Click LMB
To select Color
GGX
Christensen-Burley
Base Color
Subsurface: 0.000
Subsurface Radius
Subsurface Color
Metallic: 0.000
Specular: 0.500
Specular Tint: 0.000
Roughness: 0.500
Anisotropic: 0.000
Anisotropic Rotation: 0.000
Sheen: 0.000
Sheen Tint: 0.500
Clearcoat: 0.000
Clearcoat Roughness: 0.030
IOR: 1.450
Transmission: 0.000
Transmission Roughness: 0.000
Emission
Alpha: 1.000
Normal
Clearcoat Normal
Tangent
Node Connection Sockets

Figure 16.47

Base Color: The Material color of the selected Object in the 3D Viewport Editor.

Subsurface: Controls how much light shines through translucent substances e.g. wax.

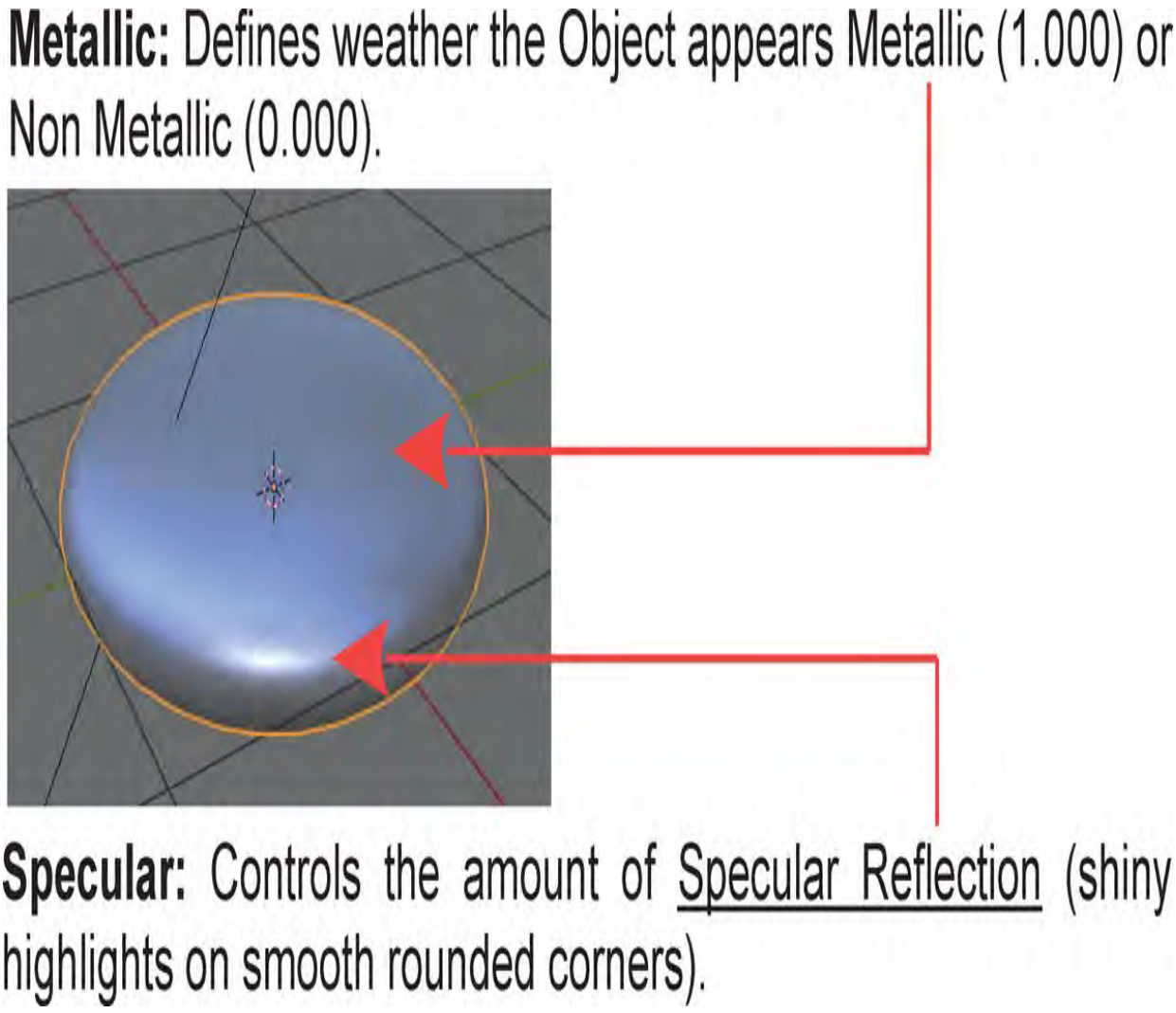


Figure 16.48

Specular Tint: Uses some of the Base Color in highlights.

Roughness: Controls how blurry or sharp reflections are.

Clearcoat: Produces extra layers of gloss, like car paint.

IOR: Transmission factor for Glass materials

With the **Principled BSDF** Node connected to the **Material Output** Node in the Shader Editor, adjusting controls (sliders) in the Node, affects the surface appearance of the selected Object in the 3D Viewport Editor.

Adding and connecting Nodes to the Principled BSDF Node was demonstrated in Figure 16.46 by the introduction of the **Color Ramp Node.**

The Color Ramp is of particular interest since it provides the means for creating a color gradient over the surface of an Object.

16.22 The Color Ramp Node

The Color Ramp is Material Node which implements a color gradient over the surface of a Mesh Object. The Node may be connected to the Principled BSDF Node as shown in Figure 16.46 but may also be configured in a simple arrangement comprising a **Converter - Color Ramp Node,** a **Shader - Diffuse BSDF Node** and a **Material Output Node.**

Remember: Material Nodes produce color on the surface of an Object which is selected in the 3D Viewport Editor. To see Material Color the 3D Viewport Editor must be in Material Preview or Rendered Viewport Shading Mode.

To add Nodes in the Shader Editor you must have a Material applied in the Properties Editor, Material Properties with Use Nodes active.

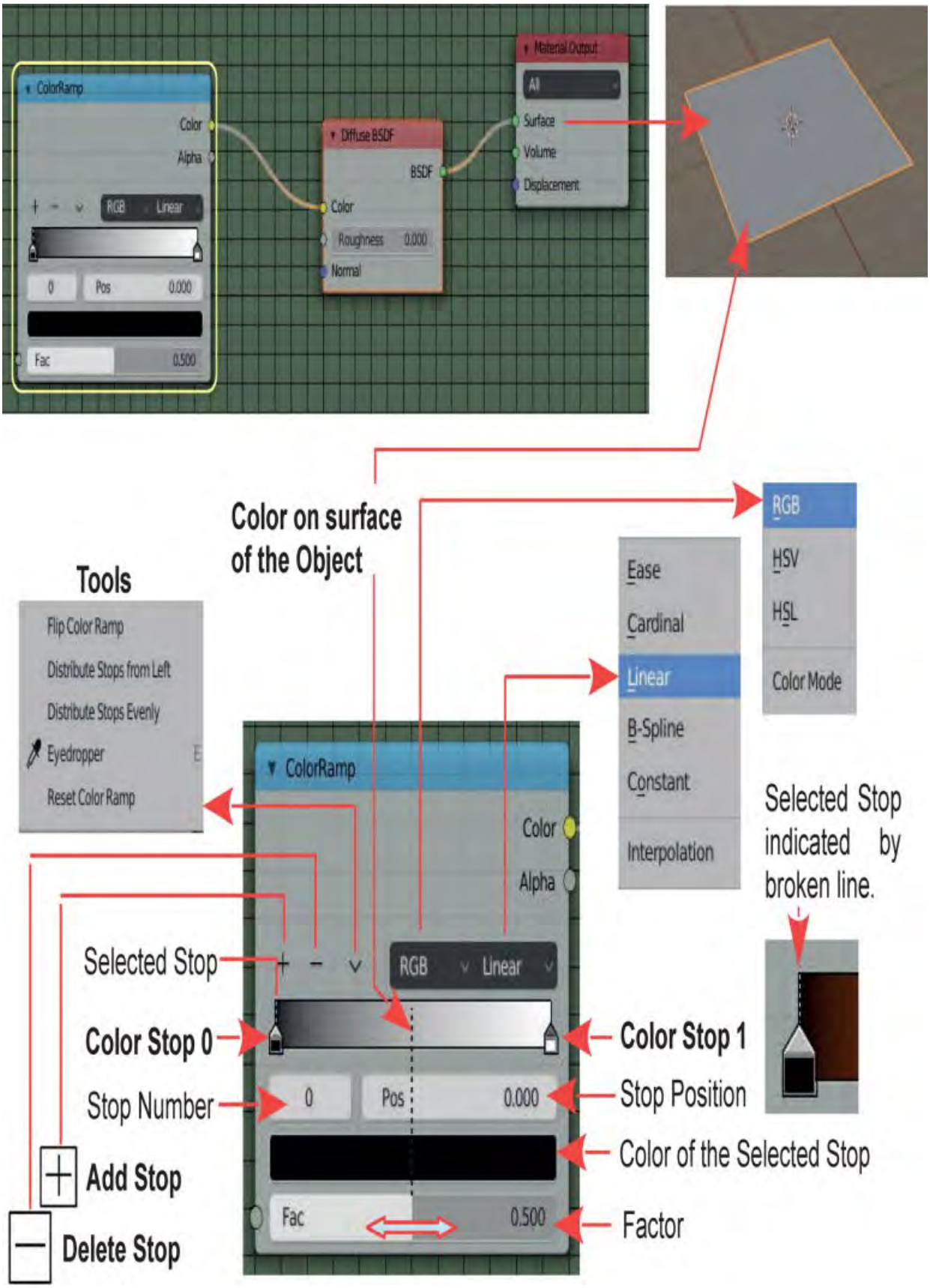

ColorRamp
Color
Alpha
RGB
Linear
0
Pos
0.000
Fac
0.500
Diffuse BSDF
BSDF
Color
Roughness
0.000
Normal
Material Output
All
Surface
Volume
Displacement
Color on surface of the Object
Tools
Flip Color Ramp
Distribute Stops from Left
Distribute Stops Evenly
Eyedropper
Reset Color Ramp
Ease
Cardinal
Linear
B-Spline
Constant
Interpolation
RGB
HSV
HSL
Color Mode
Selected Stop indicated by broken line.
Selected Stop
Color Stop 0
Color Stop 1
Stop Number
Stop Position
Color of the Selected Stop
Add Stop
Delete Stop
Factor

Figure 16.49

As you see the Color Ramp Node packs a considerable number of controls. Although not evident in the default configuration, the Node will generate a graduated Color Shading across the surface of the selected Object to which the Node is applied.

You see the shading, being applied to the Plane Object, in the **Color Ramp bar** in the center of the Node between **Color Stop 0** (Black) and **Color Stop 1** (White).

Note: The Node itself does not create the color gradient shown in the Color Ramp bar. Additional Nodes have to be connected to Map the gradient to the surface.

As it stands, the color of the Objects surface is being set by the **Factor Slider.** With the Slider mid way between **Stop 0** and **Stop 1** (Fac: 0.500) the surface color is the gray color half way along the Color Ramp Bar. Adjusting the Factor Slider alters the color between Black and White.

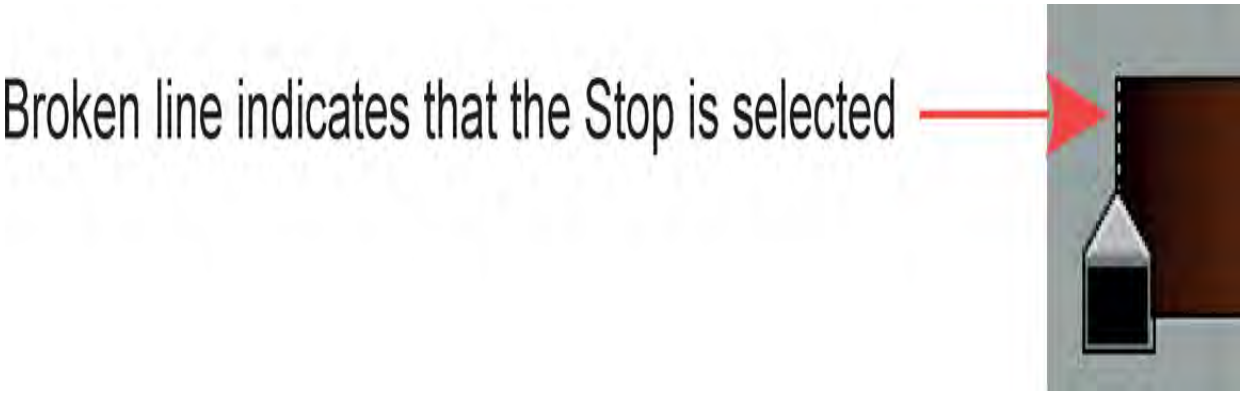


Figure 16.50

With a Color Stop selected, as indicated by the broken line above the Stop, you click in the color bar to set the Material Color **at the Stop position** in the Gradient bar.

Add a New Color Stop

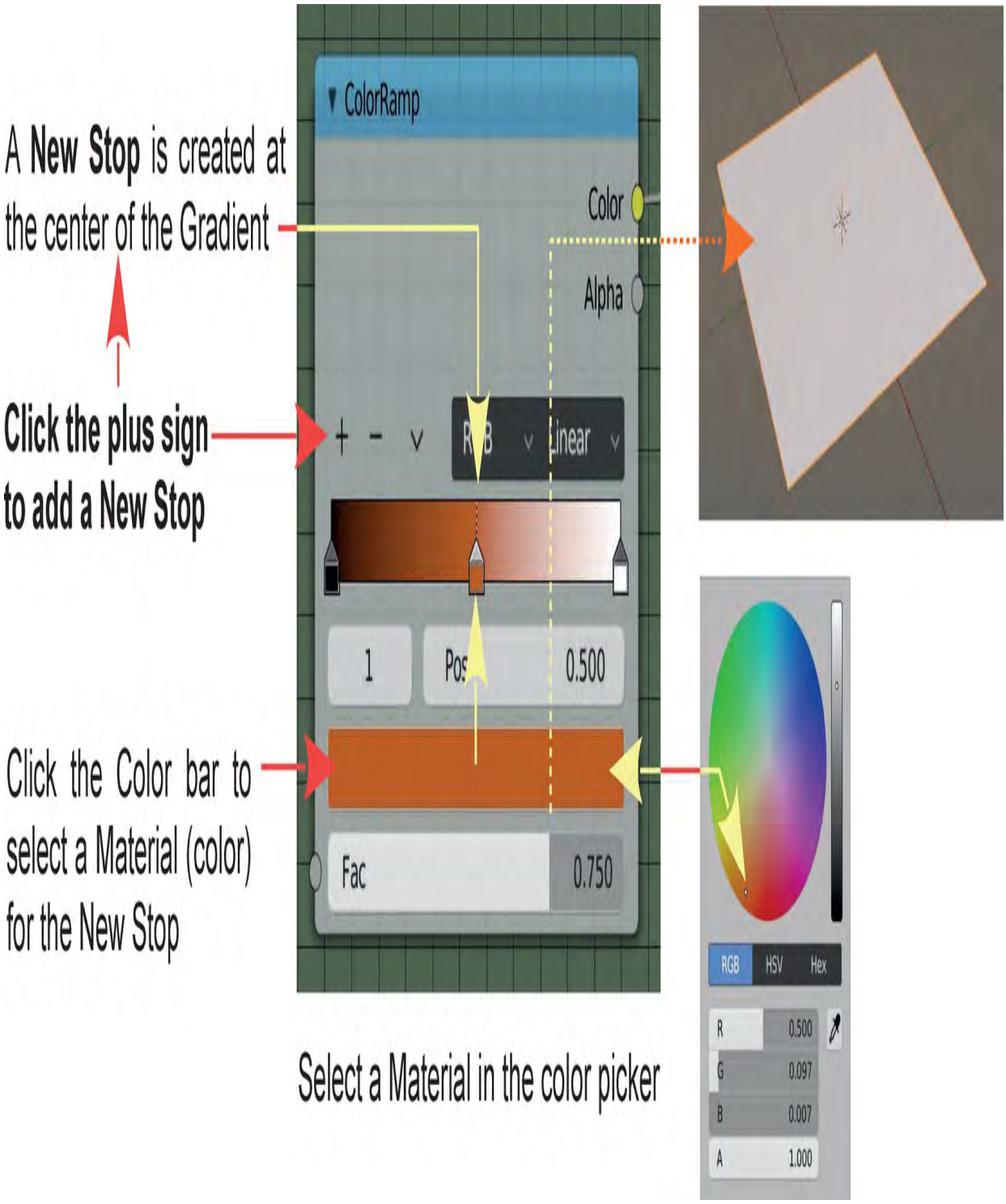


Figure 16.51

Remember: The Color Ramp Node is connected to the Diffuse BSDF Node which in turn is connected to the Material Output Node (Figure 16.49).

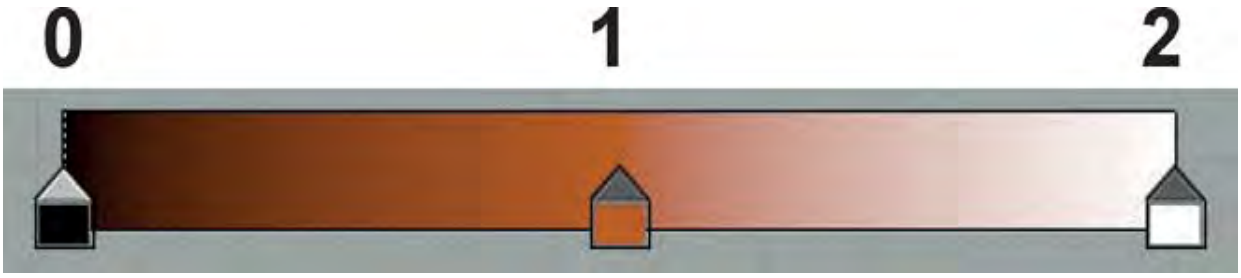


Figure 16.52

With a New Stop added, the Stops are renumbered from left to right. The Material at Stop 0 is Black, the material at Stop 1 (the New Stop) is the color selected for the Stop and the material for Stop 2 is White. The Material gradient is, therefore, from Black to Stop 1 material, to White.

Note: With the Factor Slider positioned at 0.750 the Material at that location in the gradient is displayed on the selected Object in the 3D Viewport Editor.

You may click, hold and drag any Stop left or right in the gradient bar to reposition Materials. The Material displayed on the surface of the Object will be determined by the position of the factor Slider.

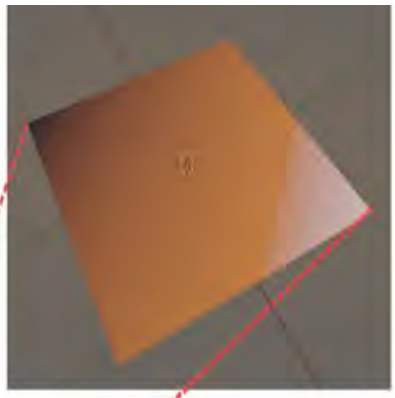
Apply the Color Gradient

To apply the Material (color) gradient to the surface of the selected Object you add a Texture Coordinate Node and a Mapping Node to the Node configuration as shown in Figure 16.53.

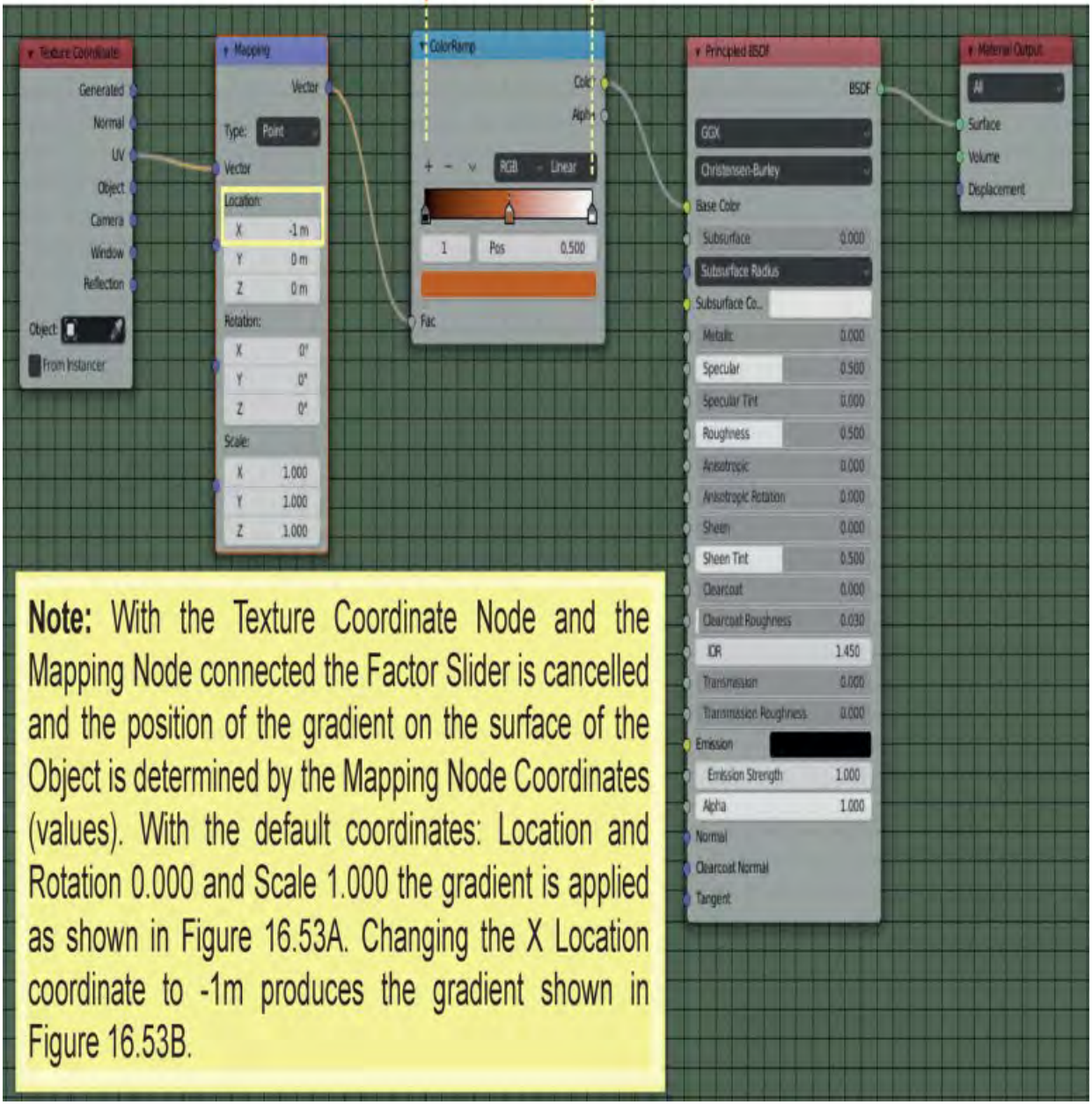
Material Preview or Rendered Viewport Shading Mode

Note: You may use the **Diffuse BSDF Node** in place of the **Principled BSDF Node**.

Material Preview or Rendered Viewport Shading Mode



Note: You may use the **Diffuse BSDF Node** in place of the **Principled BSDF Node**.



Note: With the Texture Coordinate Node and the Mapping Node connected the Factor Slider is cancelled and the position of the gradient on the surface of the Object is determined by the Mapping Node Coordinates (values). With the default coordinates: Location and Rotation 0.000 and Scale 1.000 the gradient is applied as shown in Figure 16.53A. Changing the X Location coordinate to -1m produces the gradient shown in Figure 16.53B.

Figure 16.53

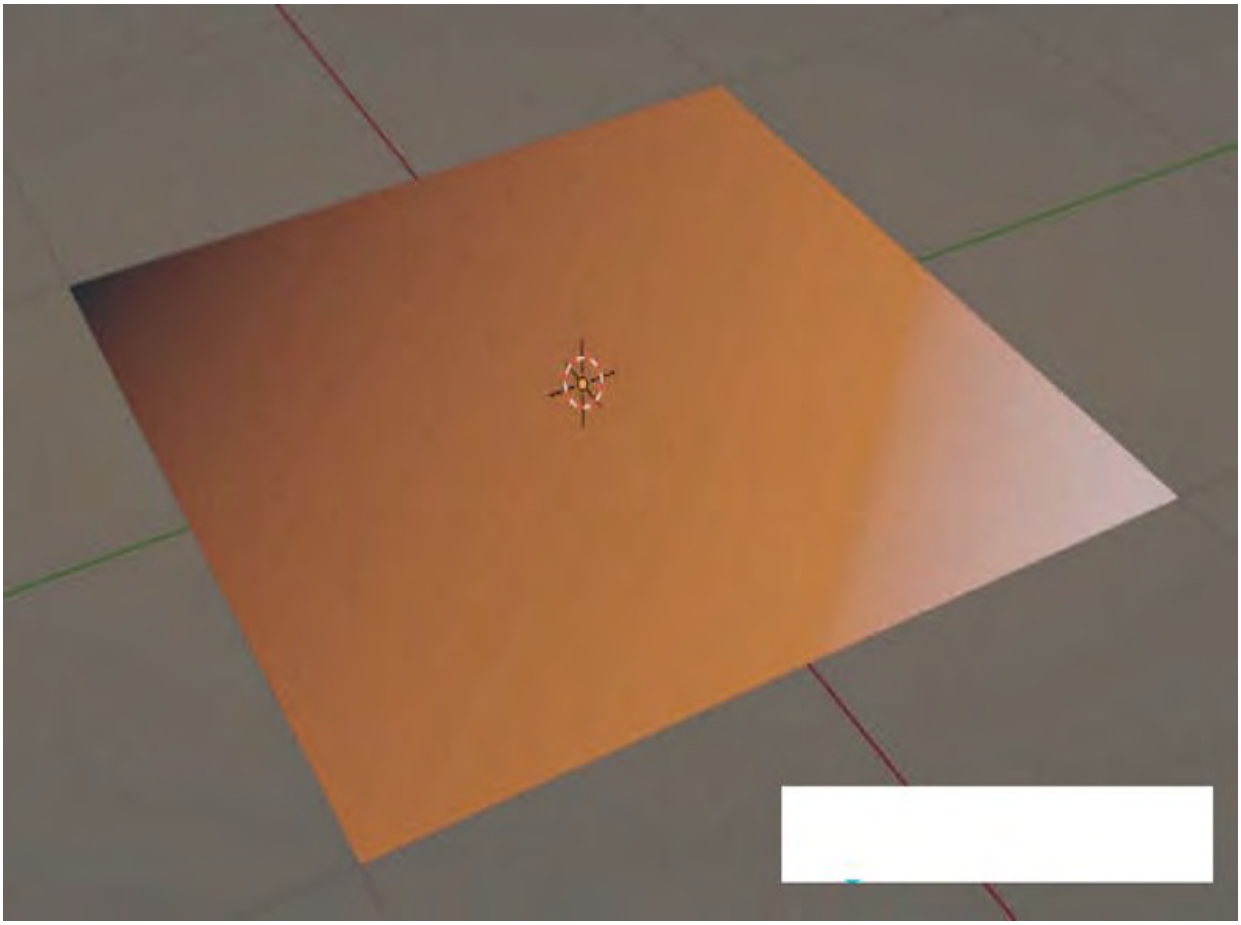


Figure 16.53A

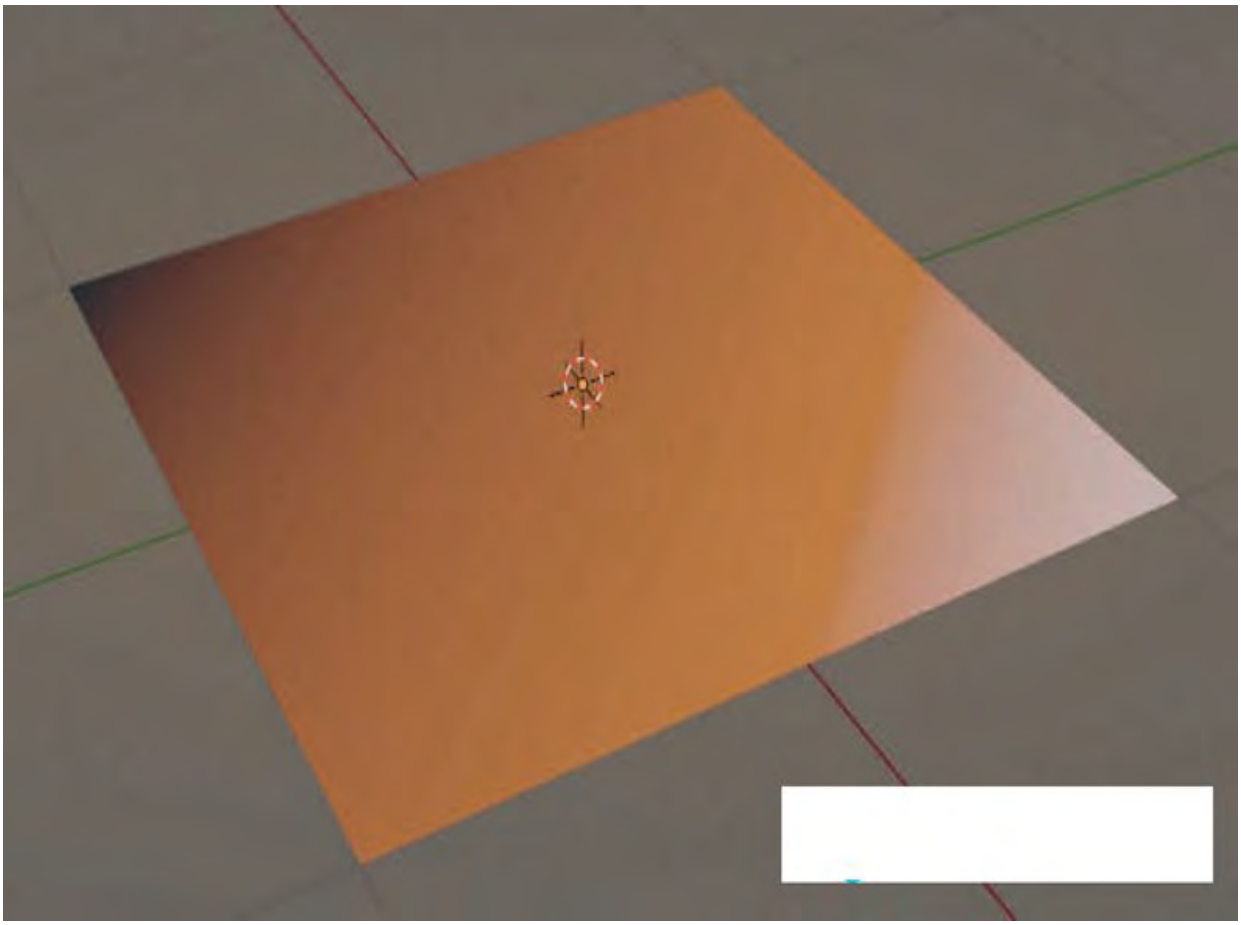
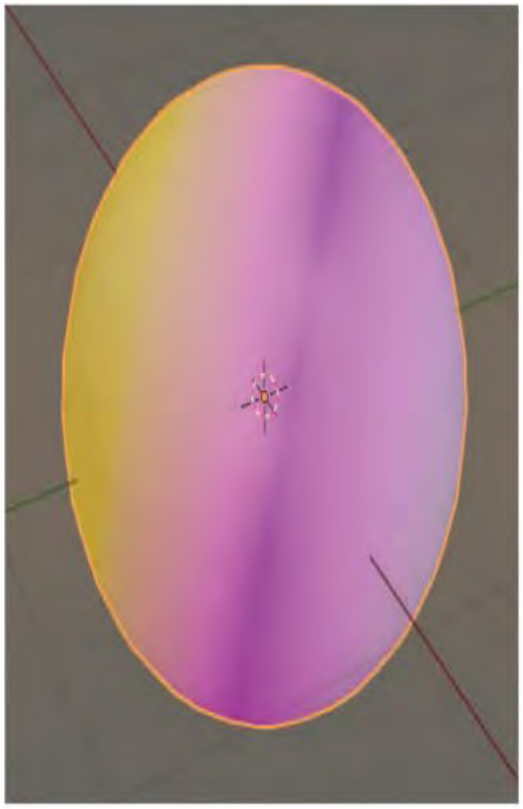
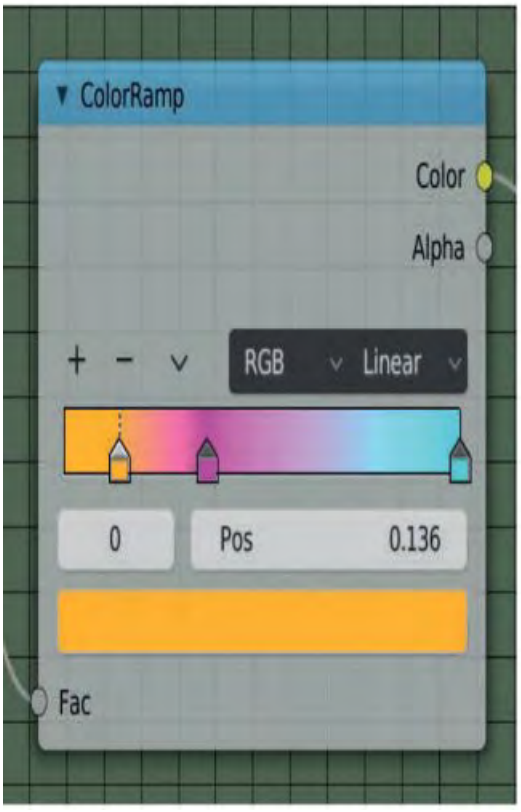


Figure 16.5BA

Gradient Examples




ColorRamp
Color
Alpha
RGB
Linear
0
Pos
0.136
Fac

ColorRamp
Color
Alpha
RGB
Linear
1
Pos
0.500
Fac

Figure 16.54

16.23 Saving Screen Space

To save Screen Space with complicated Node Systems, click on a Node to select, right click on a the Node and select **Hide** from the menu that displays to minimise a Node.

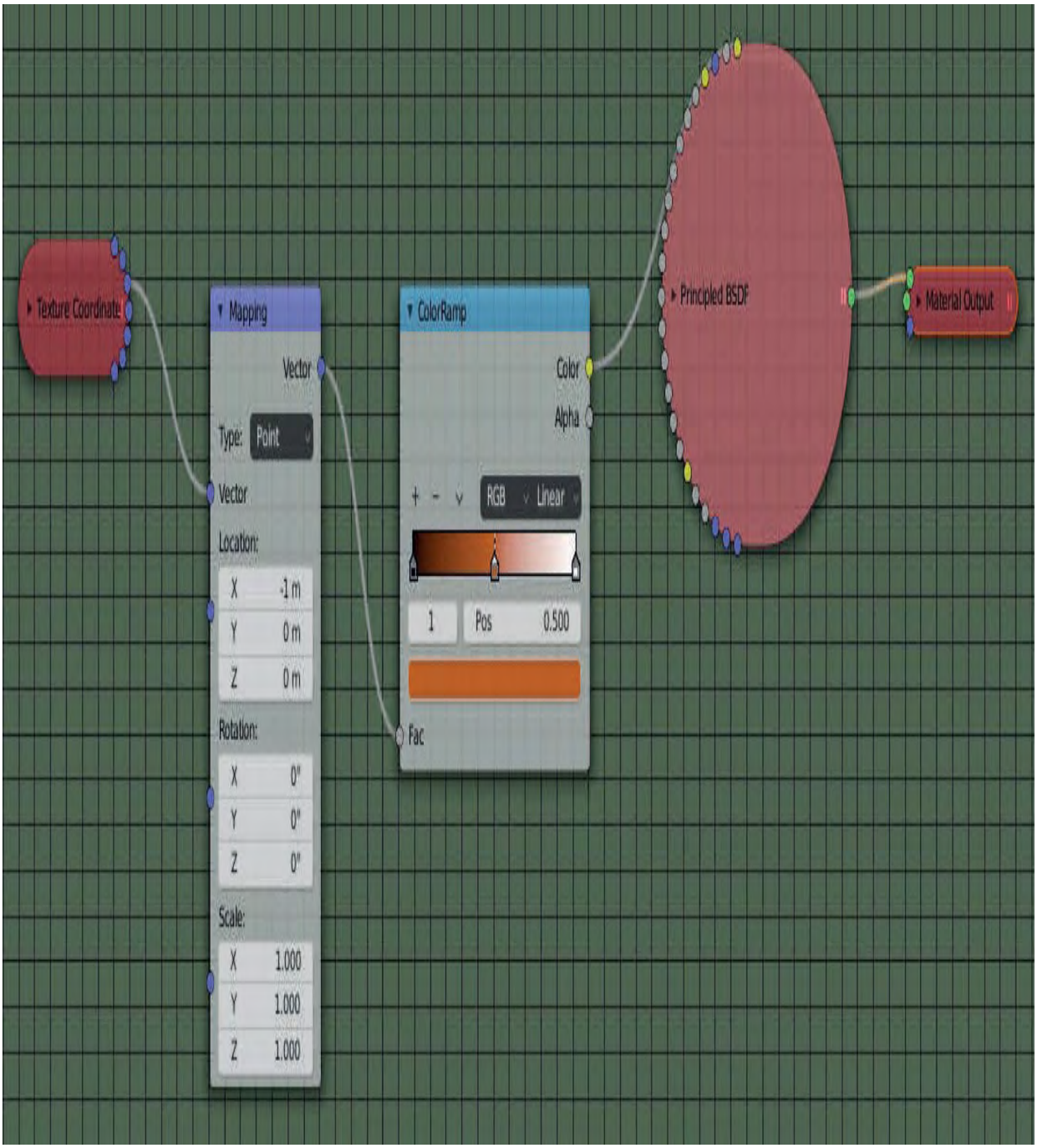


Figure 16.55

16.24 Vertex Paint

Vertex Paint allows you to manually paint a color to the surface of an Object by applying the color to the Vertices forming the Mesh Surface. Vertex

Painting is performed with a Model in the 3D Viewport Editor in **Vertex Paint Mode.**

Applying color to an Objects surface using this method is a three stage operation. The first stage is to prepare the surface, the second stage is to create Color Layers which are applied to the Model in the third stage. The Color Layers supersede any Material that has been previously applied to the Model. **Remember:** A new Object entered in the 3D Viewport Editor does **NOT** have a Material (color) applied although it displays with Blenders default gray color.

To understand the process of Vertex Painting the following exercise will apply color to **Suzanne** (the Monkey Object). You may apply color to any Object but to paint detail, the Object's Mesh Surface should contain an adequate number of Vertices. In a new Blender File delete the default Cube Object and add Suzanne.

Stage One — repare the Surface

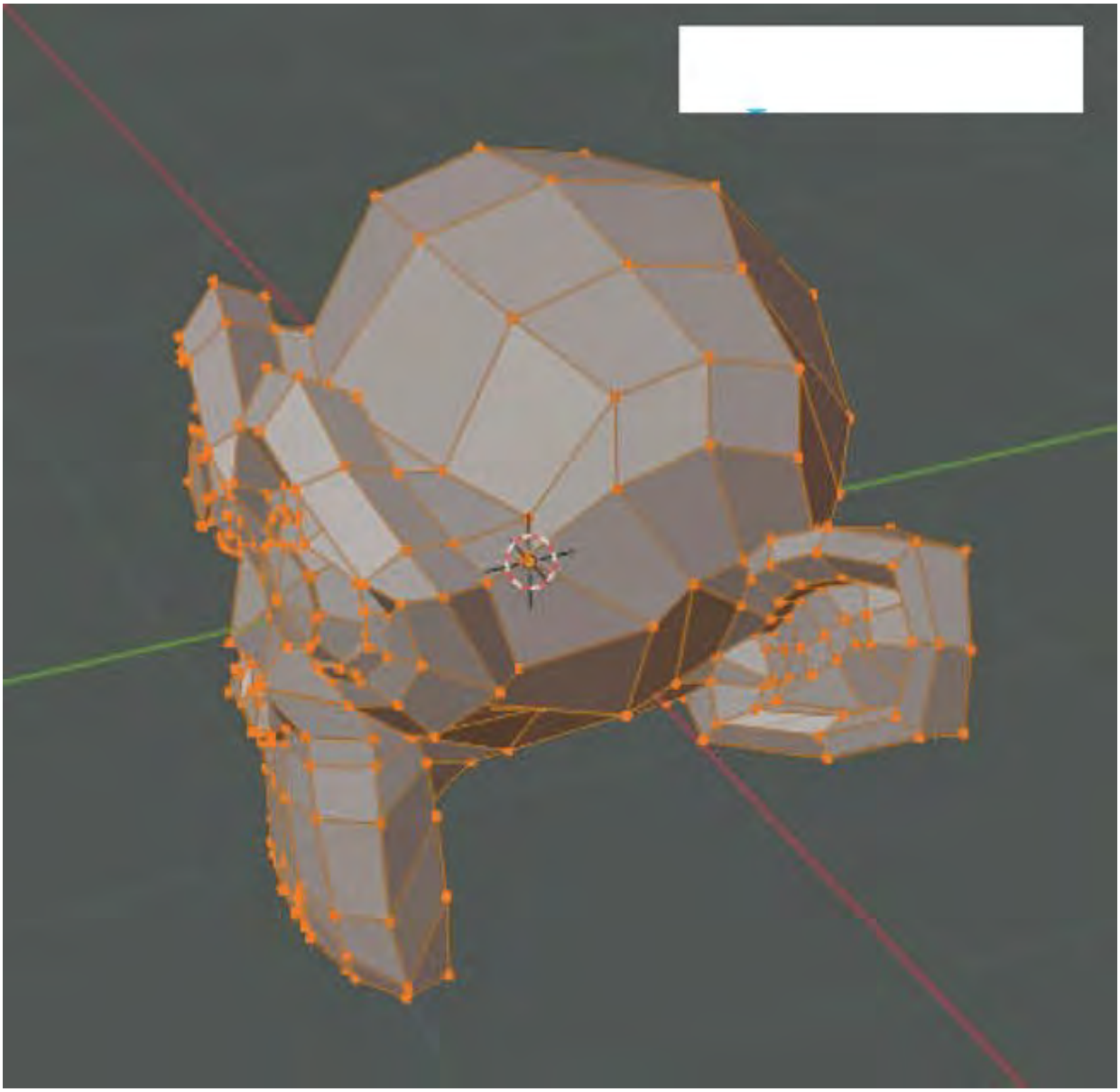


Figure 16.56

In **Edit Mode,** Suzanne displays with Vertices as shown in Figure 16.56.

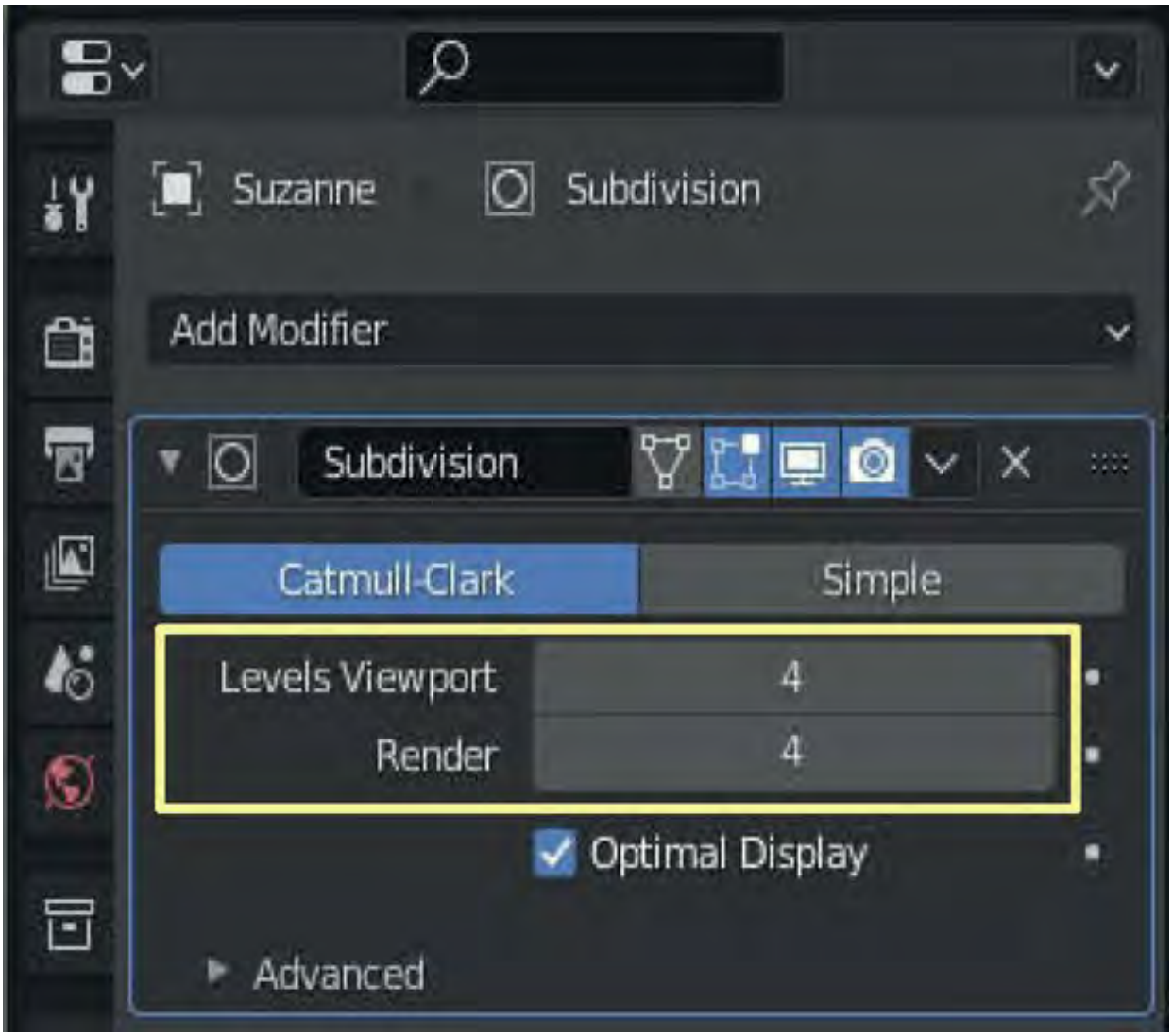


Figure 16.57

To increase the number of Vertices apply a **Subdivision Surface Modifier** to Suzanne with **Levels 4** (Figure 16.57). **Apply the Modifier in Object Mode.**

Using a **SubSerf Modifier** ensures a nice smooth surface.

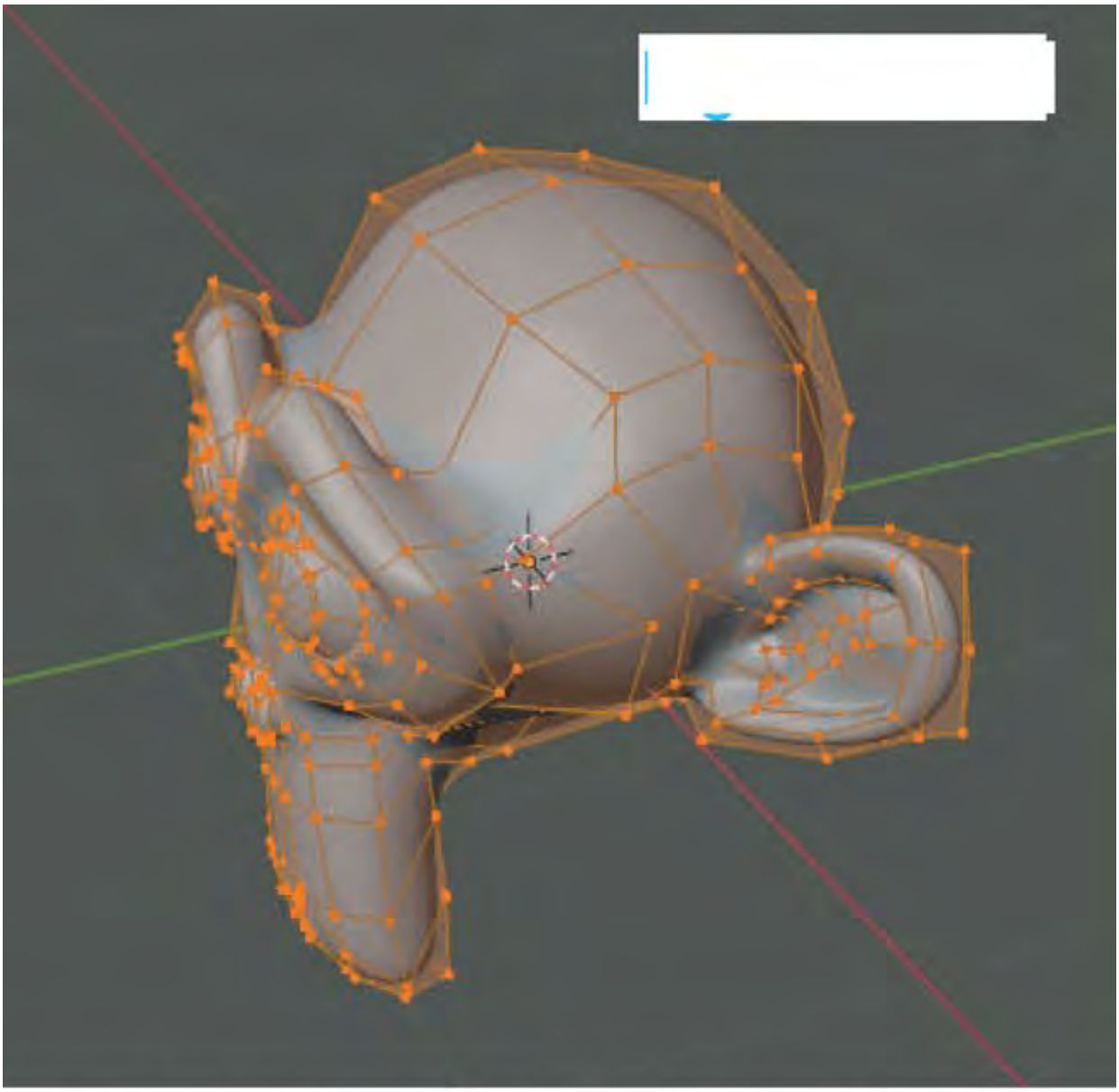


Figure 16.58

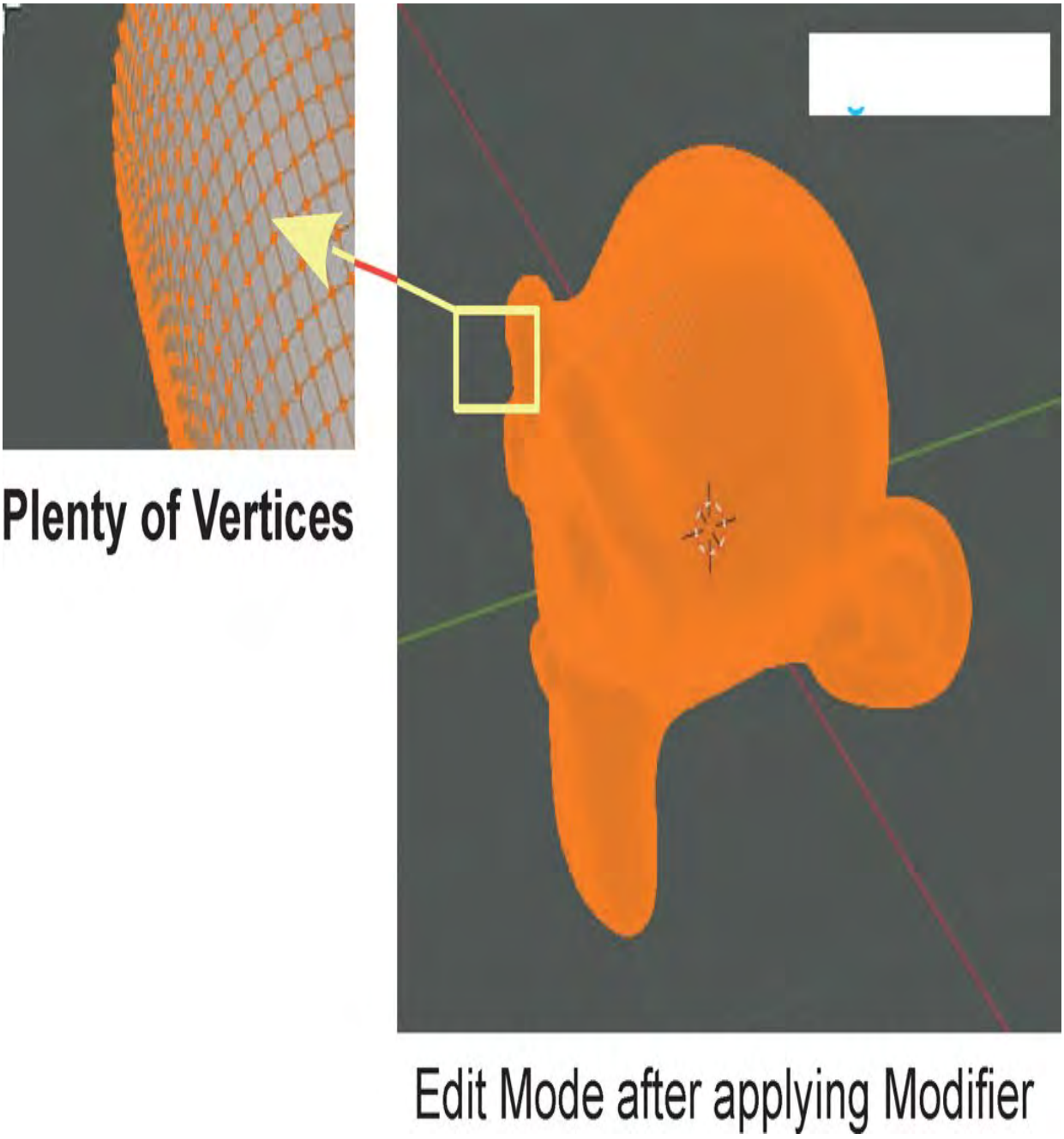


Figure 16.59

Plenty of Vertices

Vertex Paint Mode

Change the 3D Viewport Editor to Vertex Paint Mode.

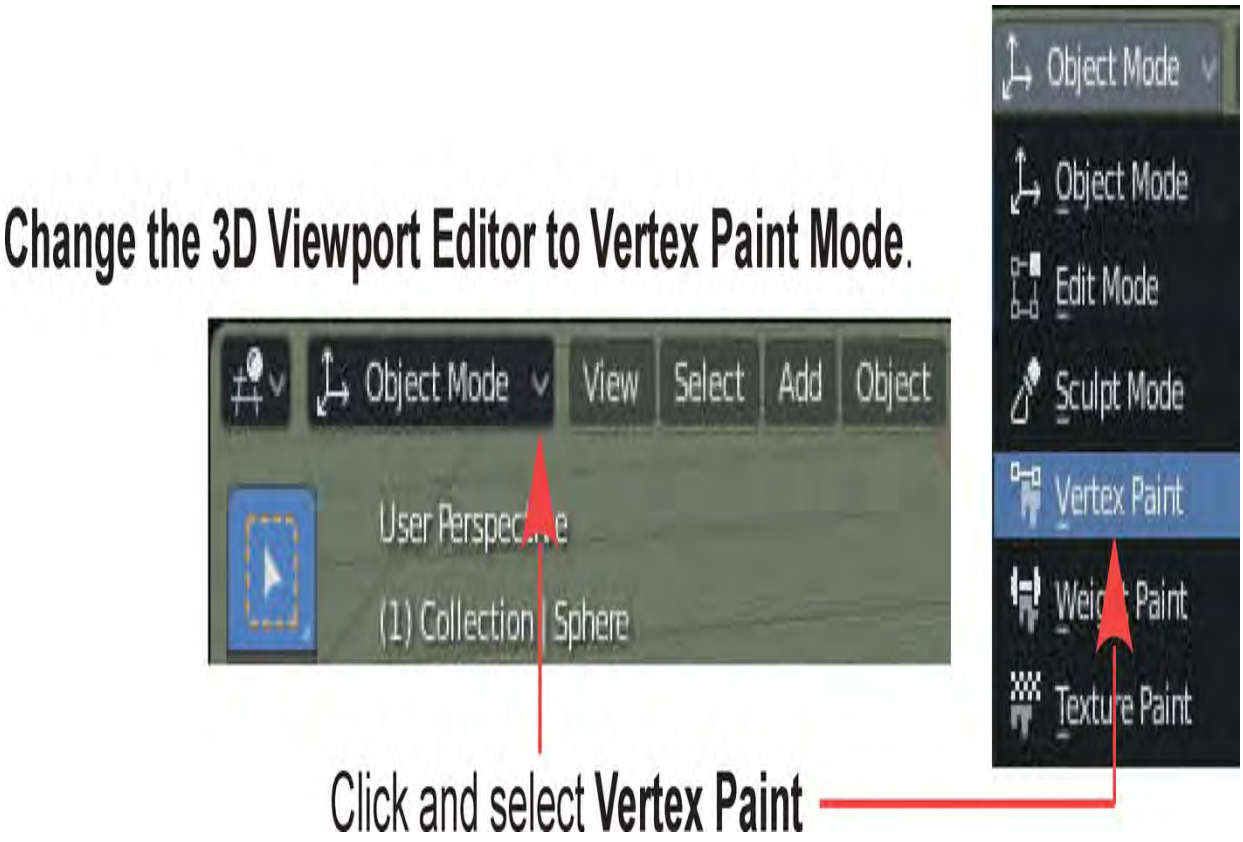


Figure 16.60

In the 3D Viewport Editor, in Vertex Paint Mode the Cursor has a circle attached (Figure 16.62). The circle is called the **Brush.** By default **Vertex Paint** is in **Draw Mode** as seen by the **Draw Tool** highlighted in the Tool Panel at the LH side of the Editor.

Brusn Gonirois

The **Bush Properties** are controlled in the **Tool Panel Information Header** which displays when **Tool Settings** is checked in the **View** Menu in the Header.

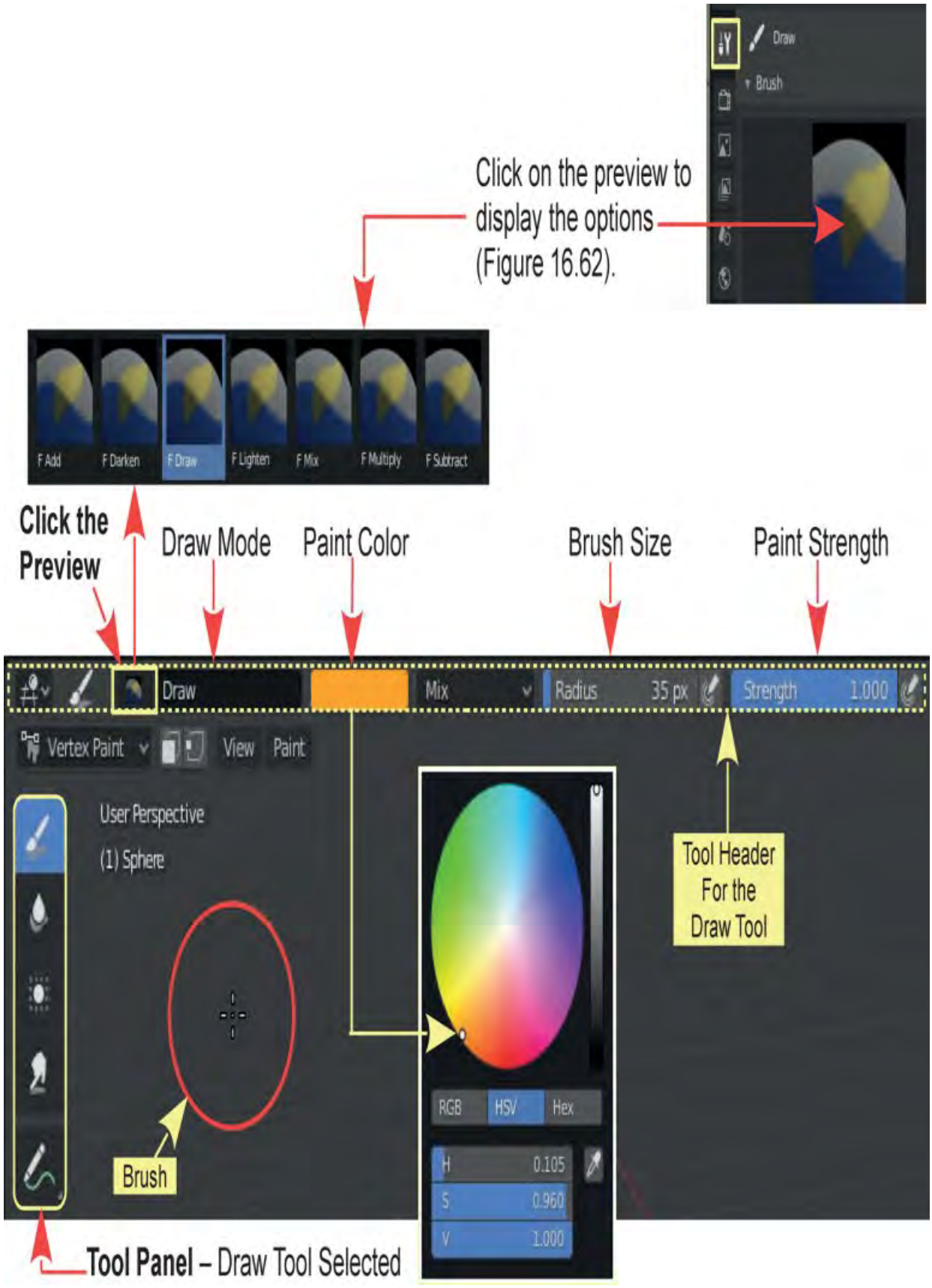

Draw
Brush
Click on the preview to display the options (Figure 16.62).
F Add
F Darken
F Draw
F Lighten
F Mix
F Multiply
F Subtract
Click the Preview
Draw Mode
Paint Color
Brush Size
Paint Strength
Draw
Mix
Radius 35 px
Strength 1.000
Vertex Paint
View
Paint
User Perspective
(1) Sphere
Tool Header For the Draw Tool
RGB
HSV
Hex
H 0.105
S 0.960
V 1.000
Brush
Tool Panel – Draw Tool Selected

Figure 16.61
Figure 16.62

Clicking on the icon to the left of **Draw** in the Header displays options for a selection of **Brush Types.** These same options are duplicated in the **Properties Editor, Active Tool and Workspace Properties** (Figure 16.62).

Note: Brush options are duplicated in the Properties Editor, Active Tools and Workspace settings

Note: The Tool Header will display slightly different controls depending on the Tool selected.

Note: Although the Brush Color Picker displays for all Tool Types it is only applicable to the draw tool and the Annotation Tool. The **Annotation Tool** allows you to freehand draw symbols and text notation to assist in construction. This notation does not render in a final image.

In the **3D Viewport Editor Header (Vertex Paint Mode)** click the color bar and select a color.

Note: You may also right click in the 3D Viewport to display the color picker with Radius and Strength values (Figure 16.63).

Painting

Figure 16.63

At this point in the exercise the 3D Viewport Editor is in Vertex Paint Mode with the Editor in **Solid Viewport Display Mode.** A Material has **NOT** been applied to Suzanne (the Model).

You may select a color, in the Header or from the Color Picker in the 3D Viewport Editor, then click, hold and drag the Brush over the surface of Suzanne (the model). The Paint Stroke will display on the surface (Figure 16.64).

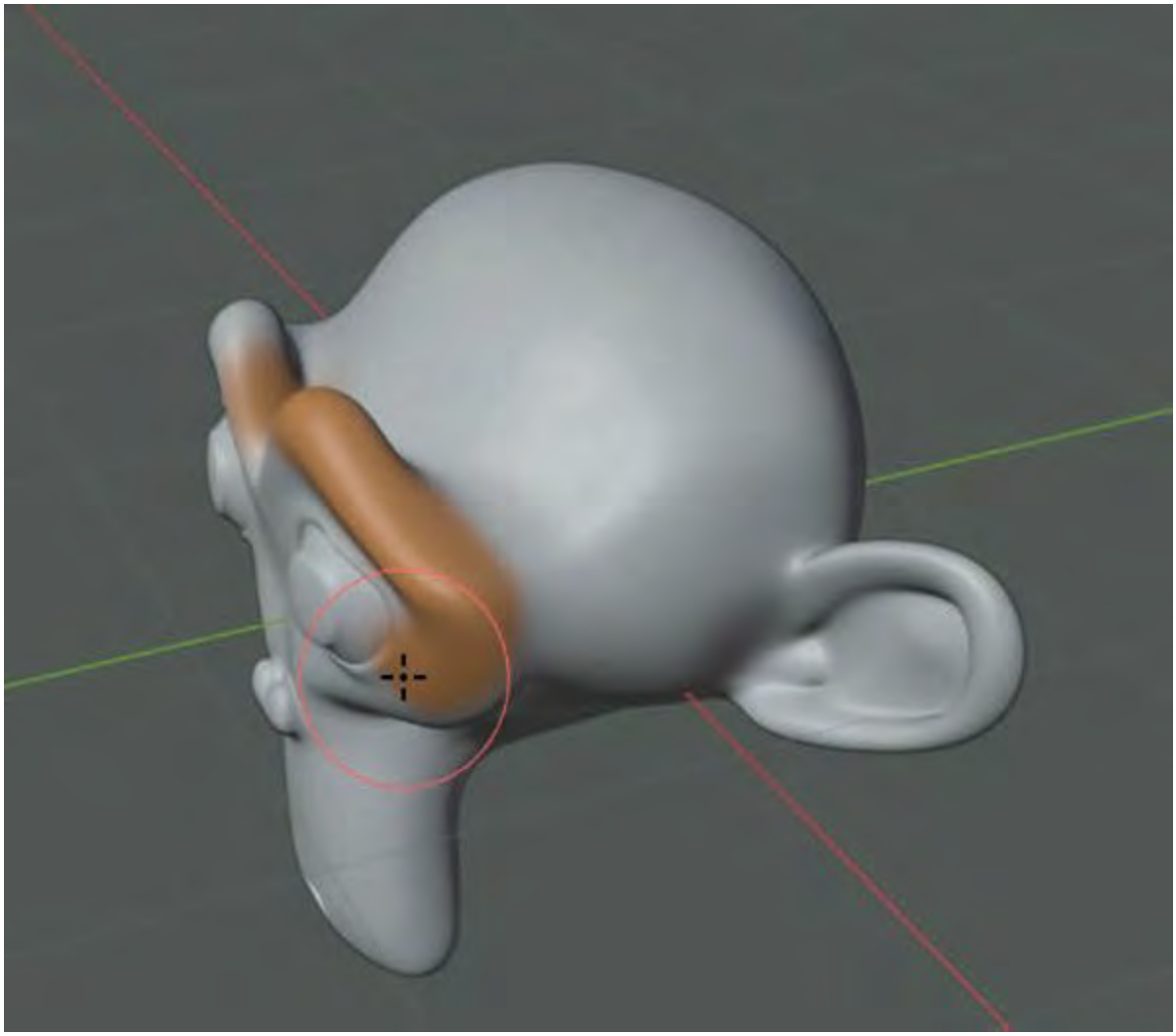


Figure 16.64

At this point it is important to understand that the color on the surface is NOT being applied to the Model and, therefore will NOT display in with the 3D Viewport Editor in Object Mode in either of the Viewport Display Modes. The color will NOT Render in an Image.

At this juncture you are generating a **Paint Layer** which will be applied to the Model in the third stage of the Vertex Paint operation.

Stage Two — Create Paint Layers.

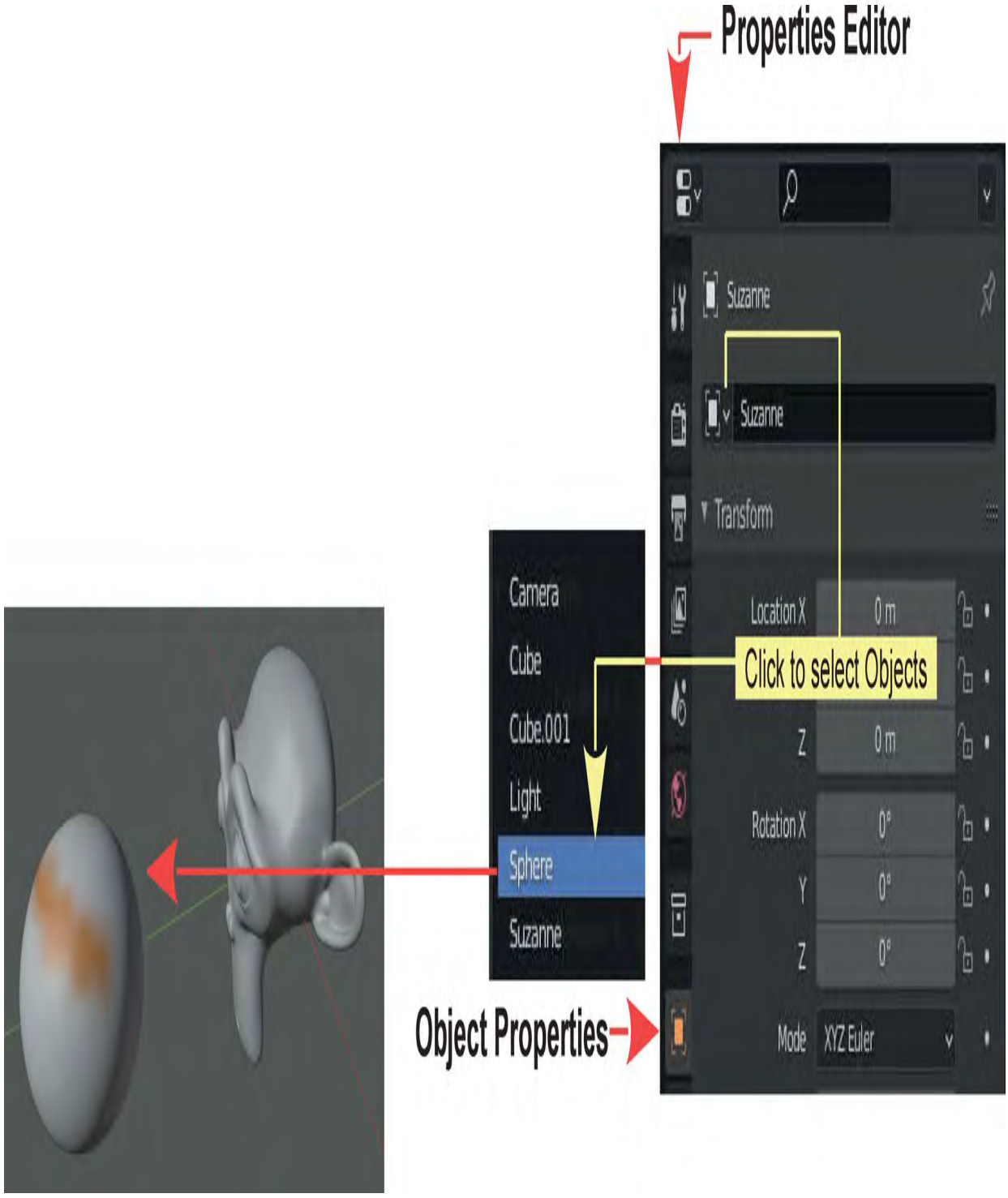


Figure 16.65

In **Object Mode, Suzanne is the selected Object,** therefore, Paint Layers will be generated for Suzanne. With a different Object in the Scene selected

Layers would be generated for that Object. While in Vertex Paint Mode you switch between Objects in the **Properties Editor, Object Properties** or in the **Outliner Editor.**

Paint Layers for Suzanne

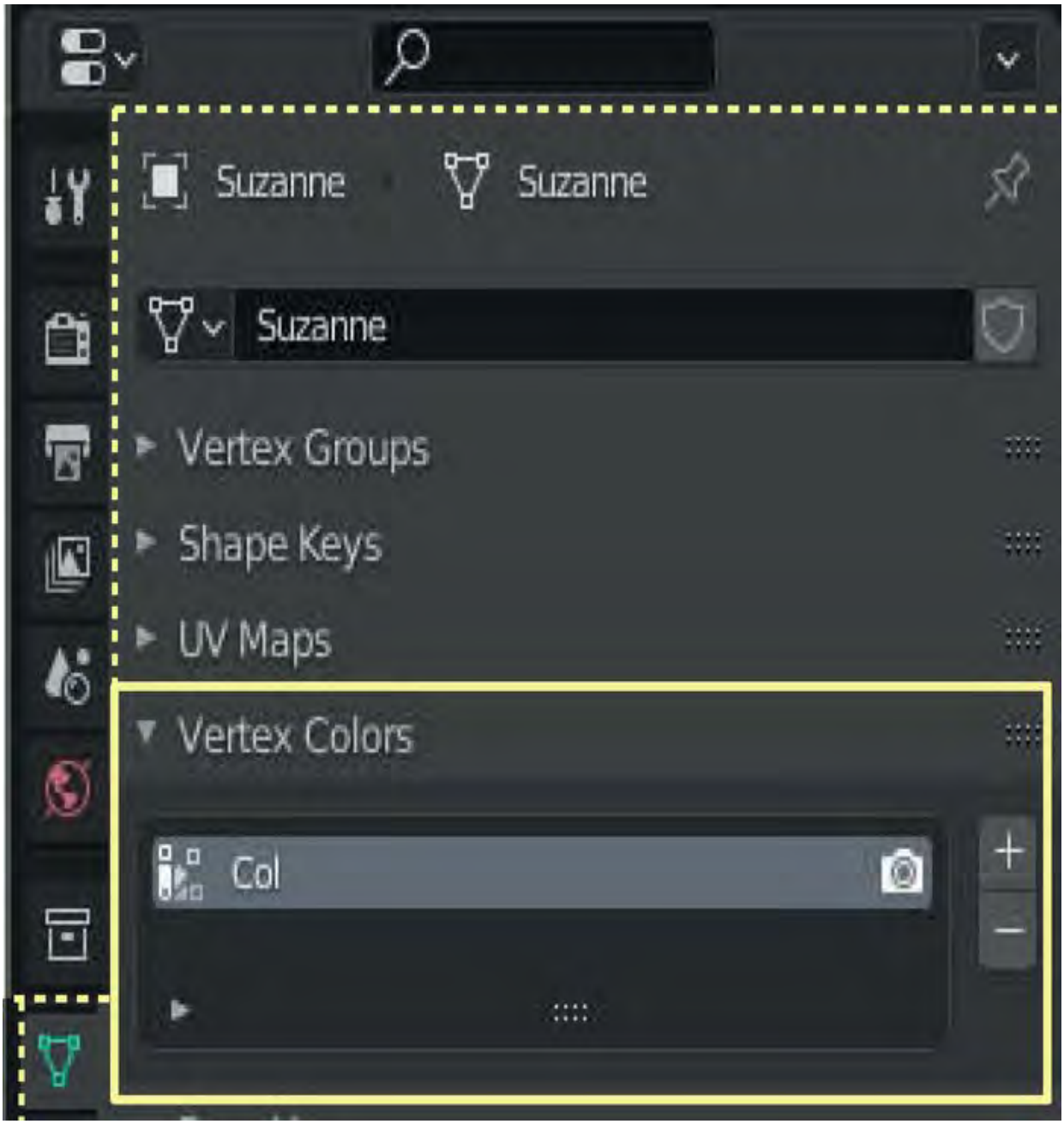


Figure 16.66

With **Suzanne** as the **Selected Object** go to the **Properties Editor, Object Data Properties, Vertex Colors Tab.**

The bar in the Vertex Colors Tab where you see **Col** is a **Vertex Paint Layer for Suzanne** (the Selected Object).

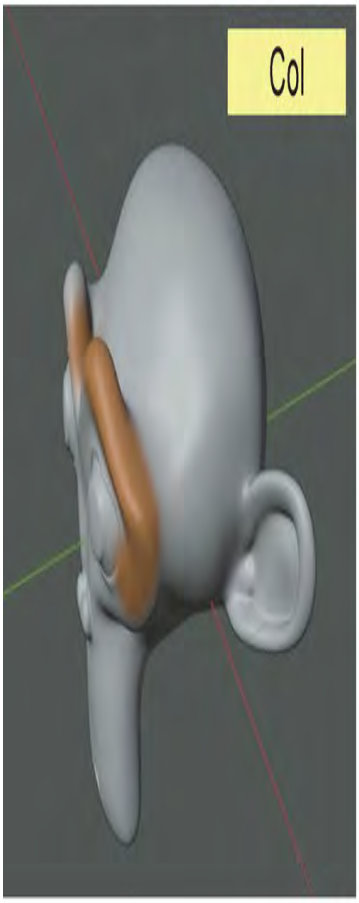
When Vertex Color is painted to the surface of Suzanne in the 3D Viewport Editor in Vertex Paint Mode, data for the color is recorded in the Layer named Col.

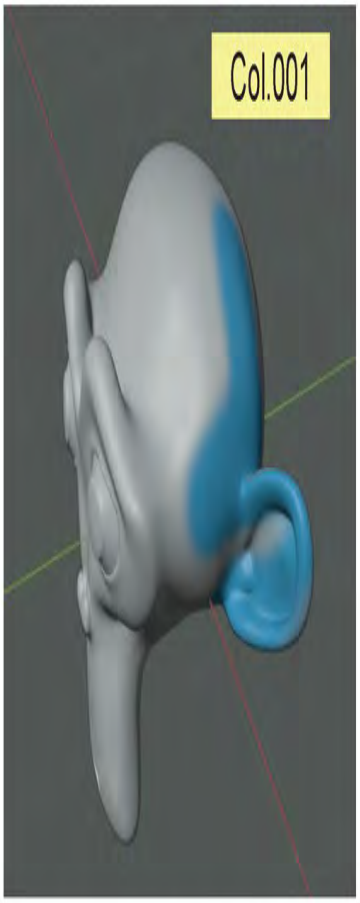
You may click the Plus sign in the Vertex Colors Tab to add a new Layer.

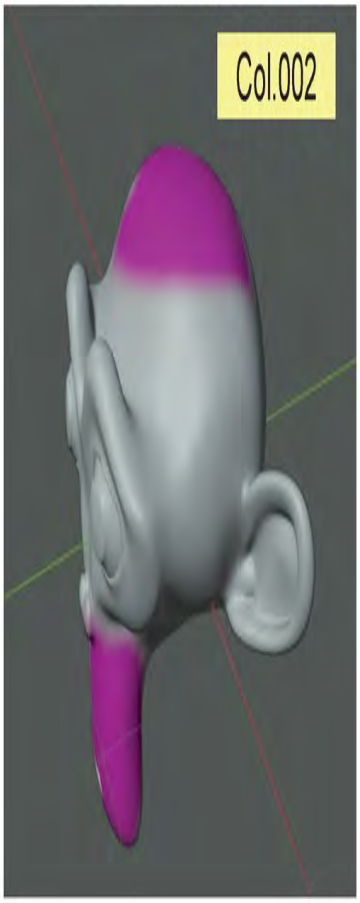
Adding a New Layer duplicates the previous Layer. To reset the new Layer to the default, select the default gray (white) in a color picker then in the **Vertex Paint Header,** click **Paint — Set Vertex Colors** or **RMB Click** in the Viewport and press **Shift + K (** applies the selected color to the Object).

Select a different color in the 3D Viewport Editor and Paint to the surface. Add a third Later and repeat the operation.

Click on each Vertex Color Layer in turn to see the three displays, Col, Co1.001, Co1.002 (Figure 16.67).







Vertex Paint Layers in the 3D Viewport Editor

Figure 16.67

Vertex Paint Layers in the 3D Viewport Editor

Remember: The 3D Viewport Editor Vertex Paint Mode is in Solid Viewport Shading Mode.

Changing the 3D Viewport Editor to Object Mode will show Suzanne in the default gray color. The Vertex Paint colors do NOT display.

To have Vertex Paint Colors display in Object Mode, Nodes in the Shader Editor have to be configured and the 3D Editor Viewport has to be in Material Preview or Rendered Viewport Shading Mode.

Change to Object Mode and have the 3D Viewport Editor in Rendered Viewport Shading Mode.

Suzanne will display in the default gray color but she is nice and smooth since the Subdivision Surface Modifier has been applied per Stage One.

Note: At this point a Material has NOT been added to Suzanne. She is showing Blender's default gray. In the Properties Editor, Material Properties go ahead and click the New button to add a Material. Suzanne is still gray but now has a Material.

Drag the upper edge of the Timeline Editor up and change it to the Shader Editor.

Stage Three — Apply the Vertex Paint in Object Mock

To apply the Vertex Paint Nodes have to be configured. In the Shader Editor you see the Principled BSDF Node connected to the Material Output Node. The base Color in the Principled BSDF Node is Blender's default gray.

Add an Input — Vertex Color Node and connect the Color Output Socket to the Base Color input of the Principled BSDF Node. **Nothing changes in the 3D Viewport Editor.**

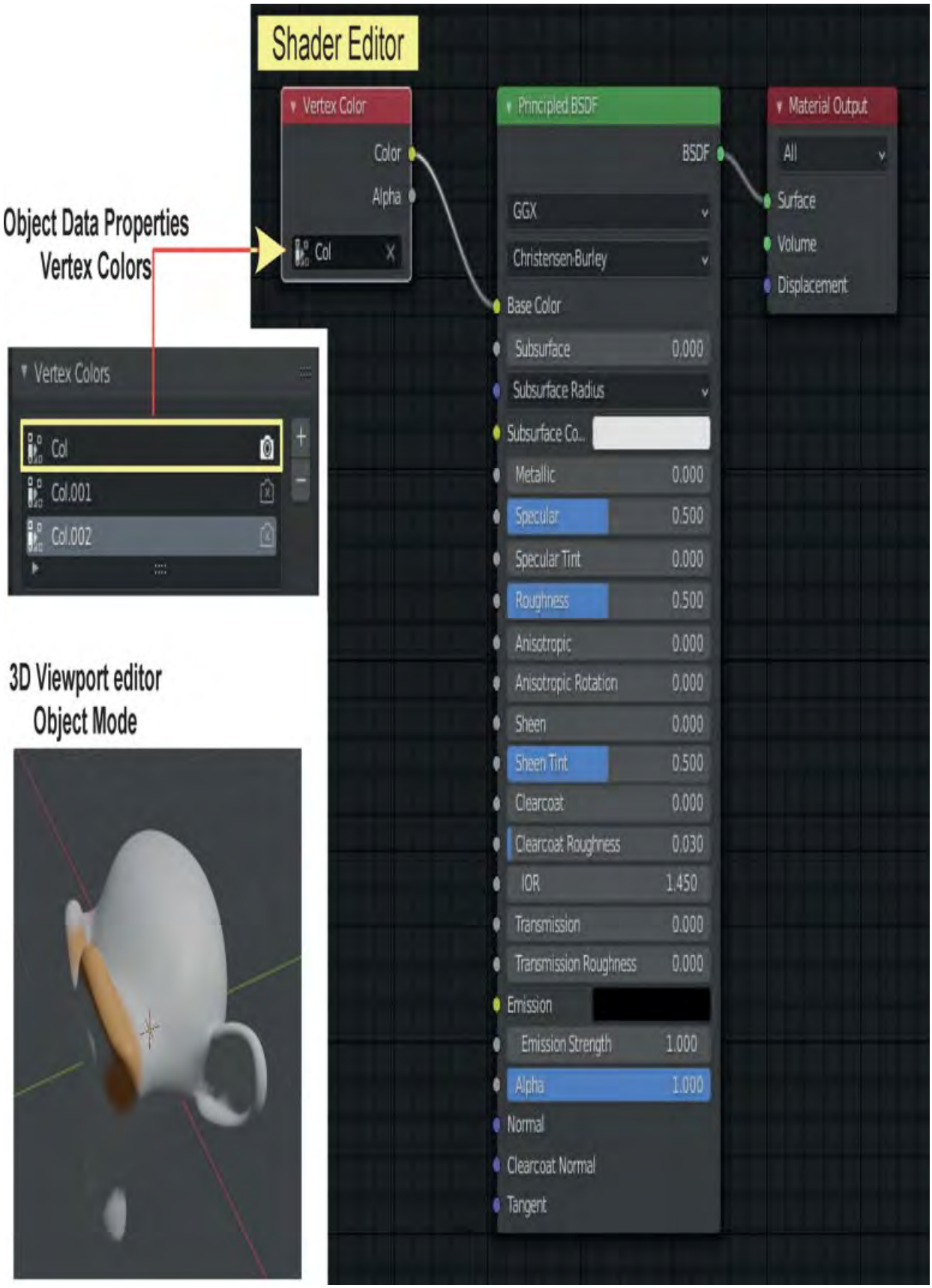

Shader Editor
Vertex Color
Color
Alpha
Col
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius
Subsurface Co...
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.500
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement
Object Data Properties
Vertex Colors
Vertex Colors
Col
Col.001
Col.002
3D Viewport editor
Object Mode

Figure 16.68

In the **Vertex Color Node** enter **Col** in the black Vertex Color bar. **Col** is the name of the first Vertex Color Layer in the Object data Properties, Vertex Colors Tab. When Vertex Color Layers have been created you can click on the black Vertex Color bar and select a Layer in the menu that displays. The Vertex Color Paint displays in the 3D Viewport. **What! It doesn't?**

Check the Viewport Shading. You have probably forgotten to change to Rendered Mode.

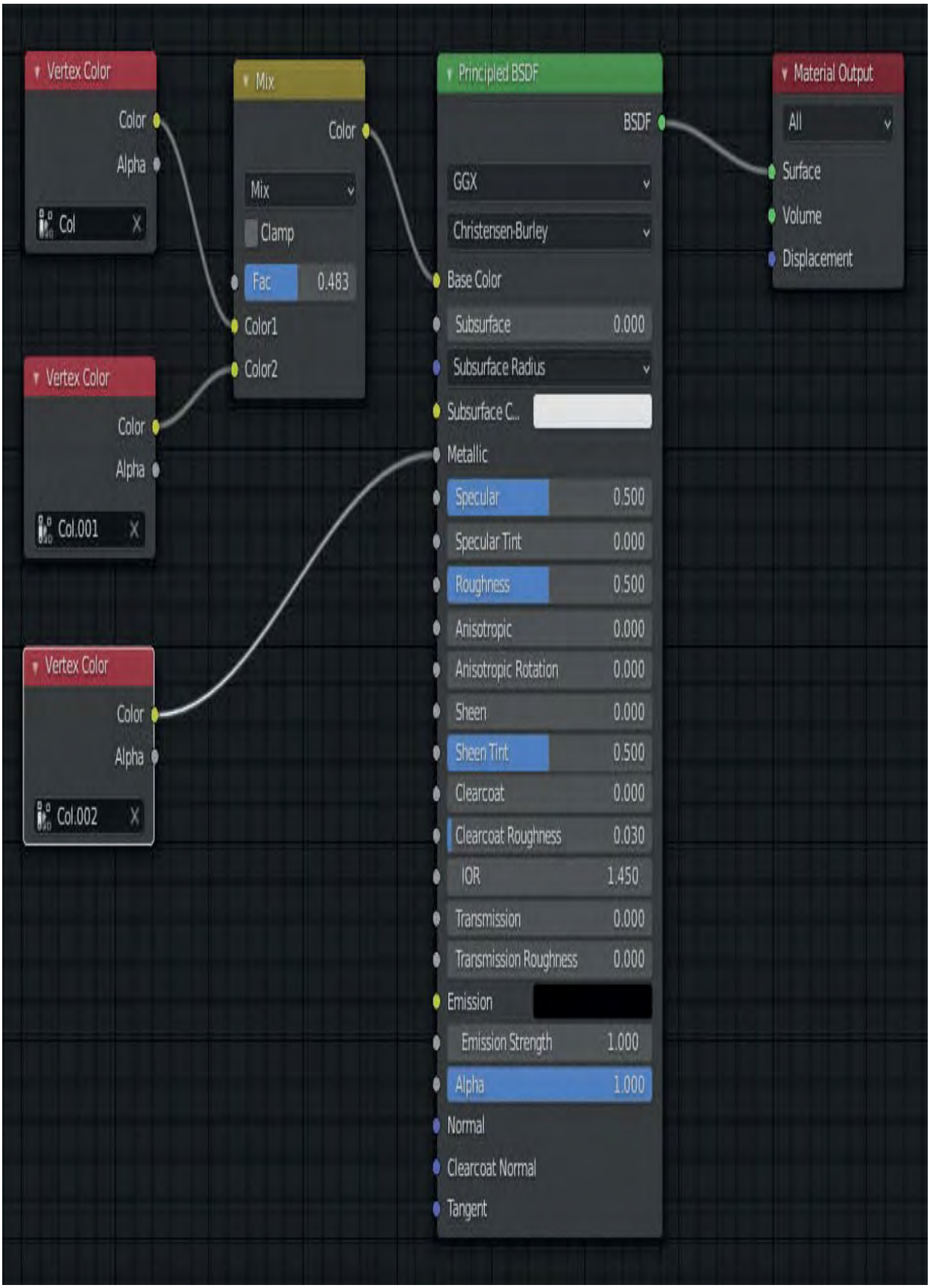

Vertex Color
Color
Alpha
Col
Mix
Color
Mix
Clamp
Fac
0.483
Color1
Color2
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
Subsurface C...
Metallic
Specular
0.500
Specular Tint
0.000
Roughness
0.500
Anisotropic
0.000
Anisotropic Rotation
0.000
Sheen
0.000
Sheen Tint
0.500
Clearcoat
0.000
Clearcoat Roughness
0.030
IOR
1.450
Transmission
0.000
Transmission Roughness
0.000
Emission
Emission Strength
1.000
Alpha
1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement
Vertex Color
Color
Alpha
Col.001
Vertex Color
Color
Alpha
Col.002

Figure 16.69

Don’t be afraid to experiment and mix things up.

The Color Palette

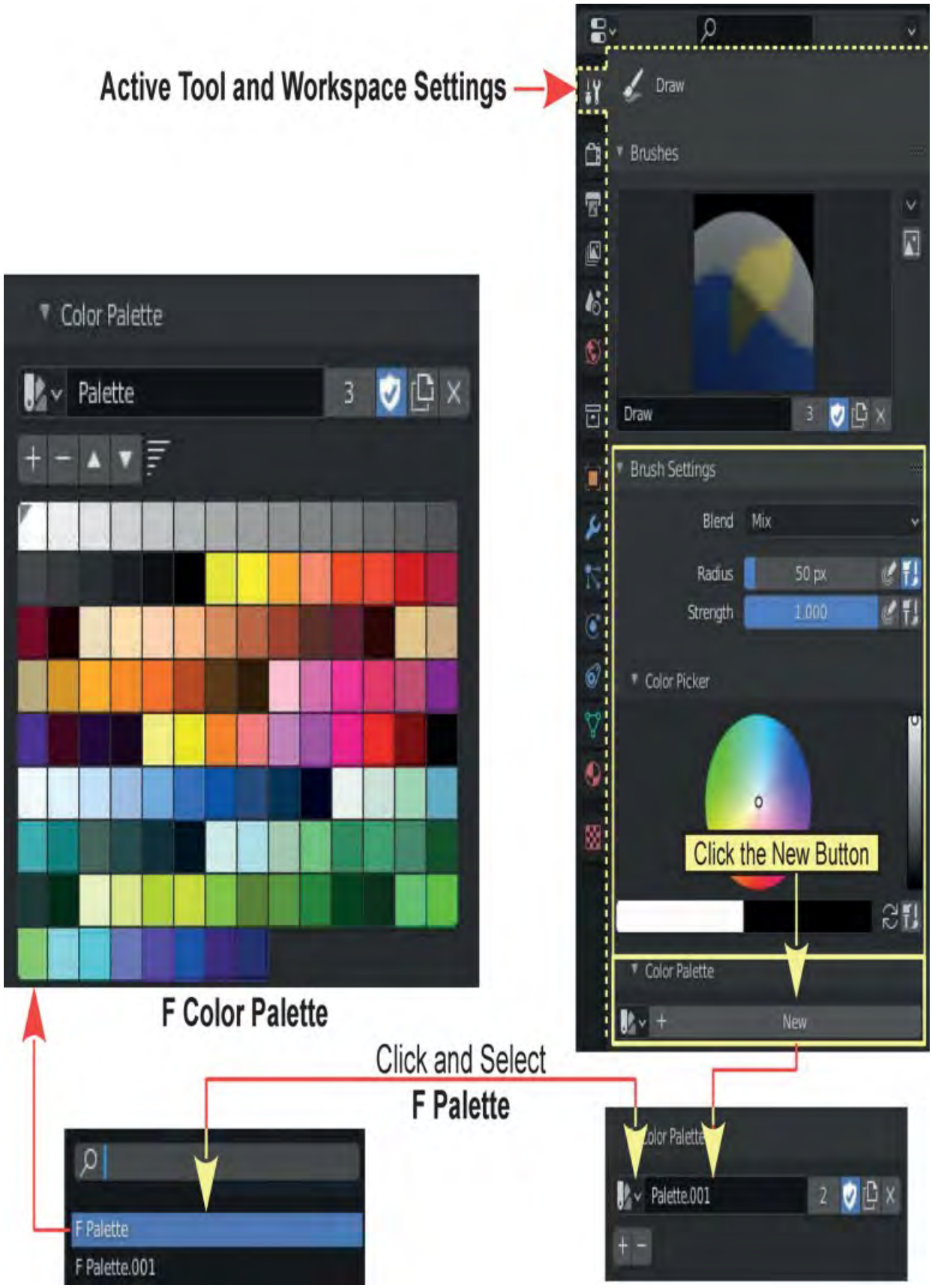

Active Tool and Workspace Settings
Draw
Brushes
Draw
3
Brush Settings
Blend
Mix
Radius
50 px
Strength
1.000
Color Picker
Click the New Button
Color Palette
New
Color Palette
Palette
3
F Color Palette
Click and Select
F Palette
F Palette
F Palette.001
Palette.001
2

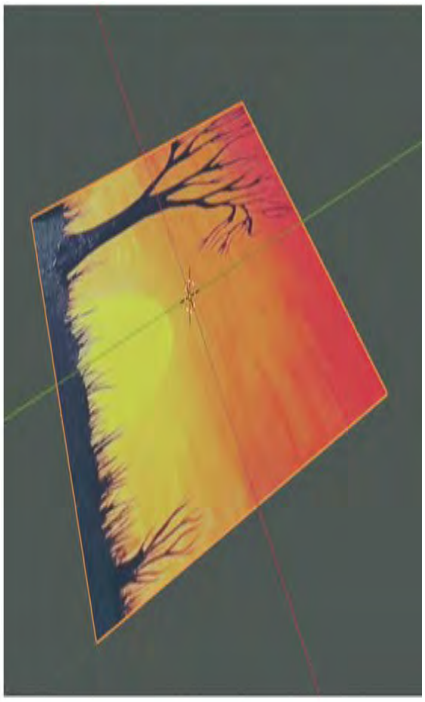
Figure 16.70

To assist when selecting colors for Vertex Paint Layers there is a **Color Palette** hidden in the Properties Editor, Active Tool and Workspace Settings, Brush Settings, Color Palette Tab.

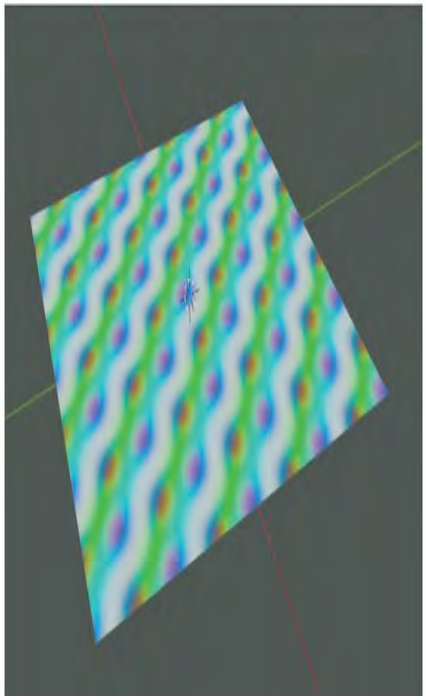
17 Textures

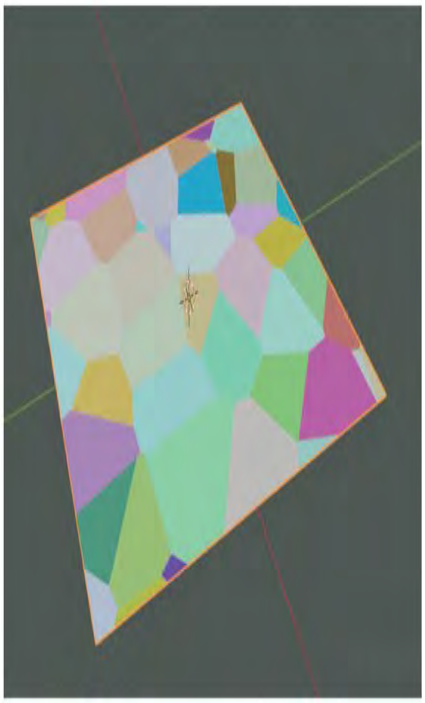
17.1 Procedural Texture
17.2 Image as a Texture
17.3 Texture Mapping
17.4 Unwrapping Options
17.5 PBR Textures
17.6 Texture Properties
17.7 Texture Painting
17.8 Texture Displacement

Textures are applied to the surface of an Object to depict the Object's composition such as whether it is glass, brick, sand, wood etc. Textures are also used to simply apply a pattern of color or shading. Textures, by and large, are images placed or mapped on to the surface and, therefore, superimpose on Material color. This type of Texture is referred to as; an **Image Texture.**



There are also **Procedural Textures** and **PBR Textures**.





Procedural Textures are patterns generated with mathematical formula and applied by computer code. PBR Textures are a Texture work-flow to simulate the lumps and bums on a surface which react with light (Physically Based Rendering). These types of Texture are special JPEG or PNG images which deform a surface and react with light adding surface lumps and bumps.

To introduce Textures they will be considered to be; **Material Textures** since they are applied to an Object's surface using the **Properties Editor, Material Properties (NOT Texture Properties)** and configured using **Nodes** in the **Shader Editor.**

17.1 Procedural Texture

To apply a Texture to a surface of an Object, have the Object (default Cube) selected in the 3D Viewport Editor. To see a Texture in the 3D Viewport Editor, on the surface of the Object, the Editor must be in **Material Preview** or **Rendered Viewport Shading** Mode.

The starting point for applying a Texture to an Object is in the **Properties Editor, Material Properties, Surface Tab.**

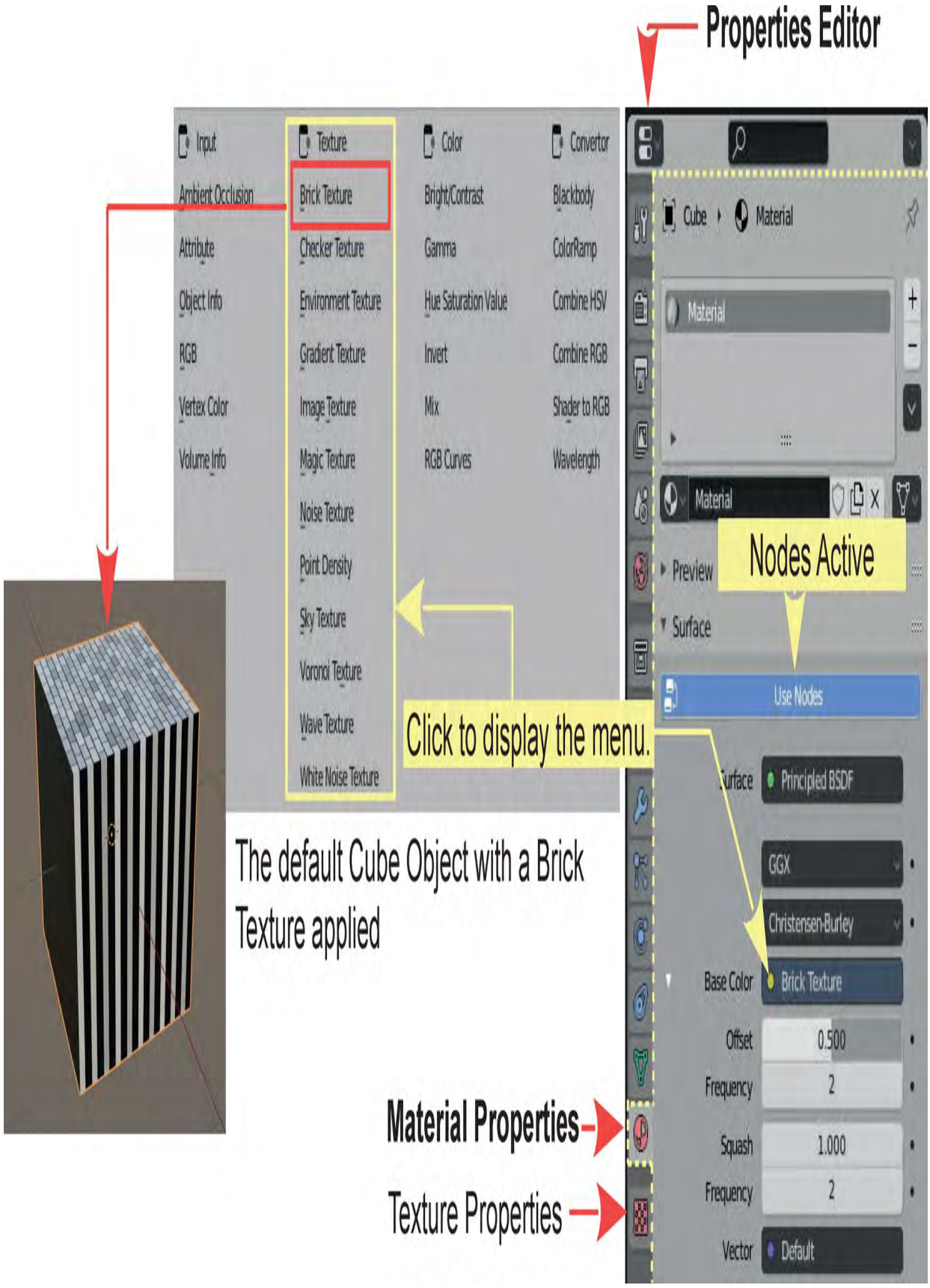

Properties Editor
Input
Ambient Occlusion
Attribute
Object Info
RGB
Vertex Color
Volume Info
Texture
Brick Texture
Checker Texture
Environment Texture
Gradient Texture
Image Texture
Magic Texture
Noise Texture
Point Density
Sky Texture
Voronoi Texture
Wave Texture
White Noise Texture
Color
Bright/Contrast
Gamma
Hue Saturation Value
Invert
Mix
RGB Curves
Convertor
Blackbody
ColorRamp
Combine HSV
Combine RGB
Shader to RGB
Wavelength
Cube
Material
Material
Material
Nodes Active
Preview
Surface
Use Nodes
Click to display the menu.
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Brick Texture
Offset
0.500
Frequency
2
Squash
1.000
Frequency
2
Vector
Default
The default Cube Object with a Brick Texture applied
Material Properties
Texture Properties

Figure 17.1

You will note, there is a **Texture Properties** button in the Properties Editor which, at this point, contains no entries. This will be explained later.

Remember; the default Cube Object has a Material pre-applied. New Objects entered into a Scene **Do NOT** have Materials until the **New Button** is pressed in the **Properties Editor, Material Properties.** Pressing the New Button applies a Material and displays Material Controls. Applying a Material to an Object in the Properties Editor, **with Nodes activated** automatically places a **Principled BSDF Node** connected the **Material Output Node** in the **Shader Editor.**

To apply a Texture, click on the dot adjacent to the **Base Color** bar in the Properties Editor, Material Properties and in the selection menu that displays select one of the Textures in the **Texture Category** (Figure 17.1).

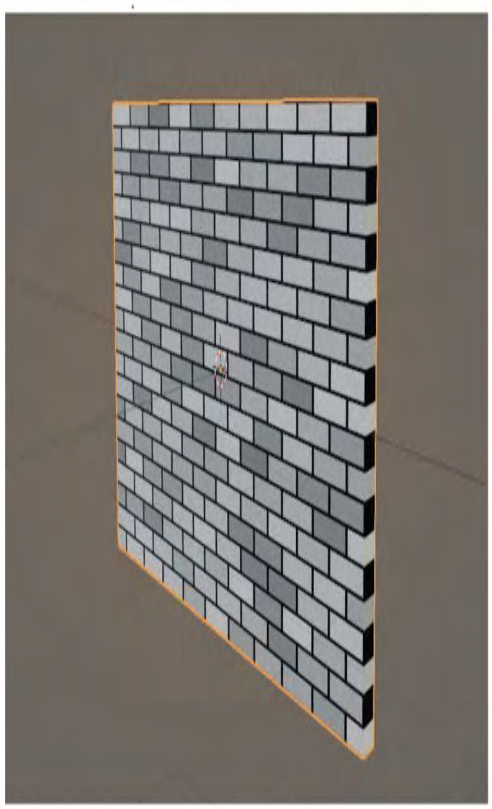
The Textures listed are called **Procedural Textures** which are built into Blender. This means the texture is produced by code which generates a pattern in accordance with mathematical formula. There is one exception in the Texture Category which is the **Image Texture.**

The Image Texture enables the use of an Image saved on the computer being used. In this case the image would be **Mapped** to the surface of the Object (explanation to follow).

Since the application of the Texture is emanating from the Material Properties the Textures will be considered **Material Textures (** Procedural and Image).

In Figure17.1 a Procedural Brick Texture is applied to the default Cube Object in the 3D Viewport Editor. When the **Shader Editor** is opened you see the **Node arrangement** creating the Texture.

The default Cube Object rotated and scaled to produce a brick wall.



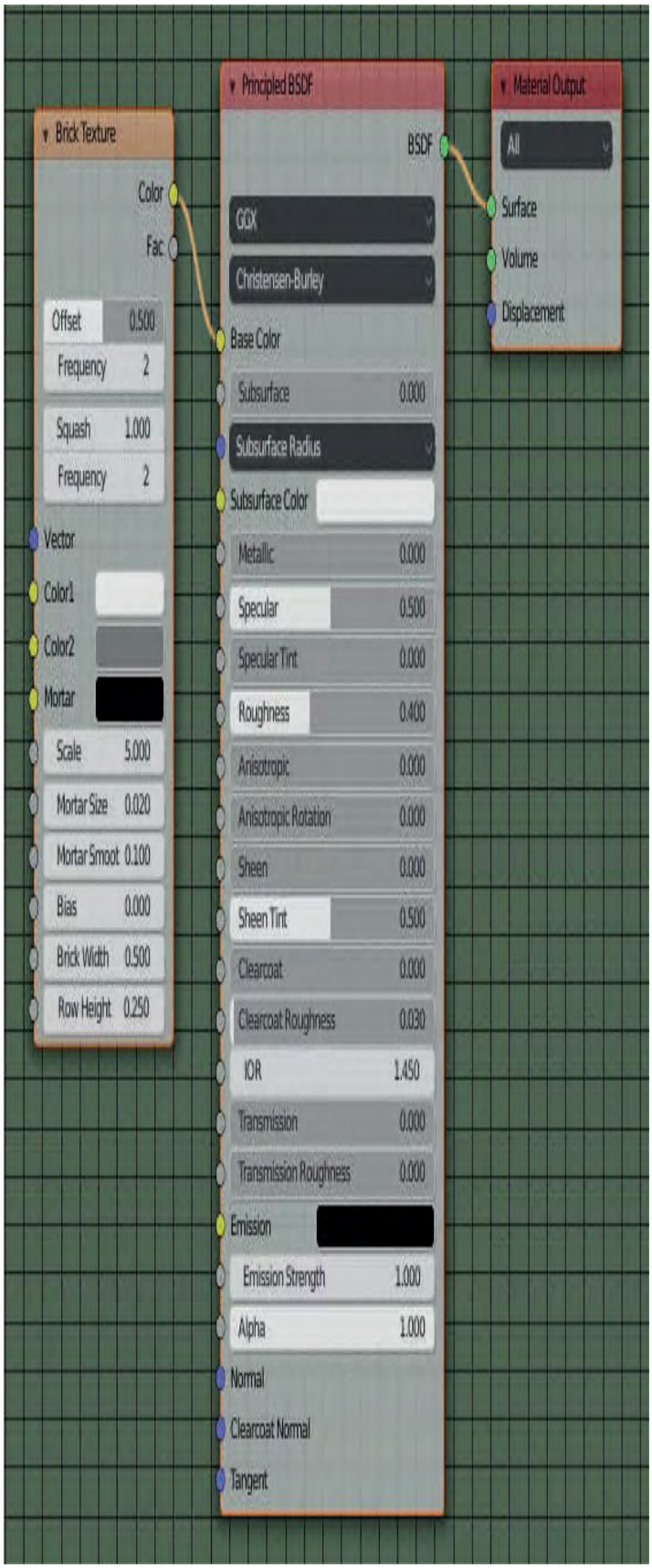

Brick Texture
Color
Fac
Offset 0.500
Frequency 2
Squash 1.000
Frequency 2
Vector
Color1
Color2
Mortar
Scale 5.000
Mortar Size 0.020
Mortar Smoot 0.100
Bias 0.000
Brick Width 0.500
Row Height 0.250
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius
Subsurface Color
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.400
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement

Figure 17.2

Values may be adjusted in the Brick Texture Node to alter the Texture Output.

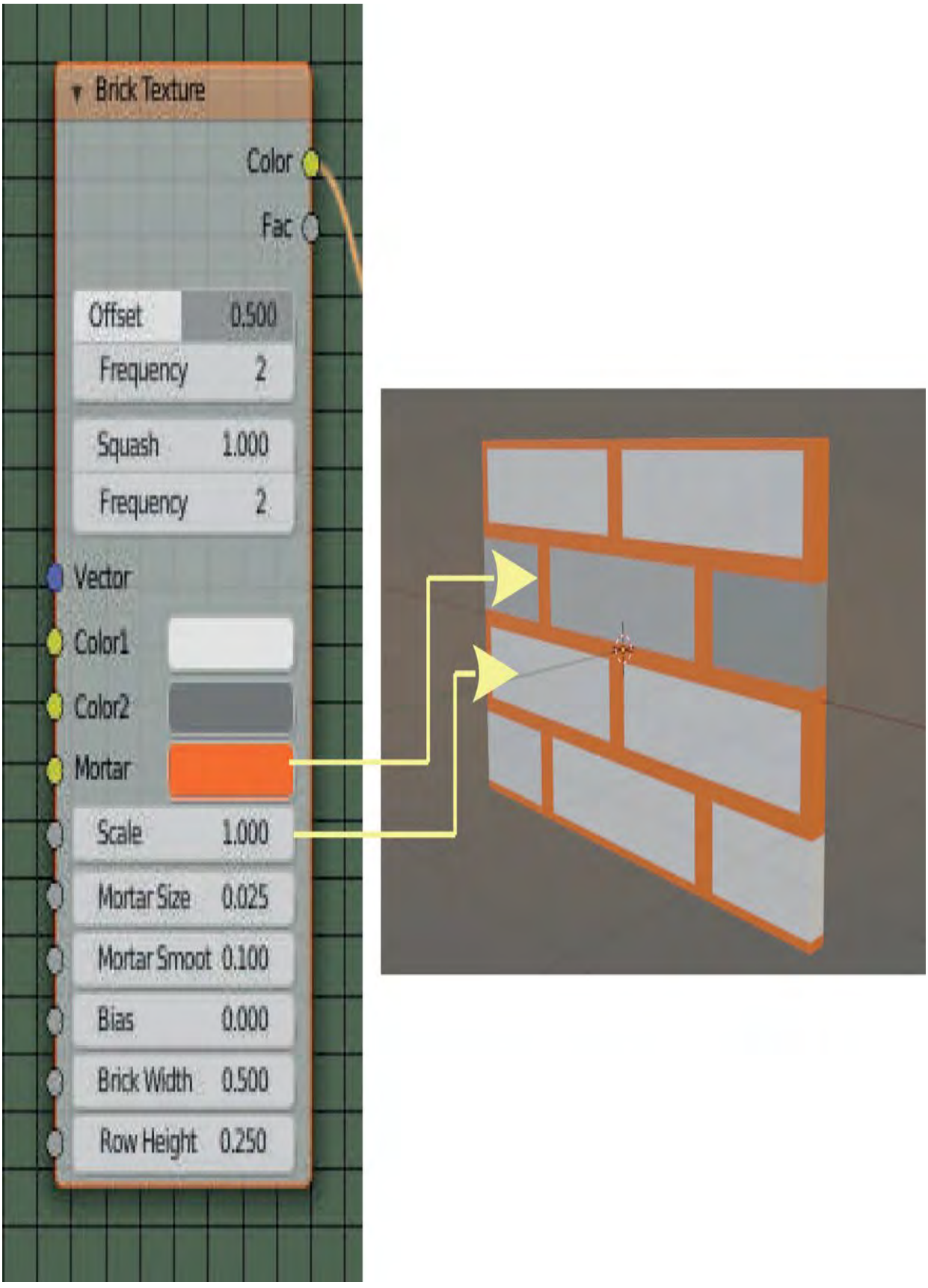

Brick Texture
Color
Fac
Offset 0.500
Frequency 2
Squash 1.000
Frequency 2
Vector
Color1
Color2
Mortar
Scale 1.000
Mortar Size 0.025
Mortar Smoot 0.100
Bias 0.000
Brick Width 0.500
Row Height 0.250

Figure 17.3

The size of the Wall is unaltered. Only the Scale of the bricks and mortar change.

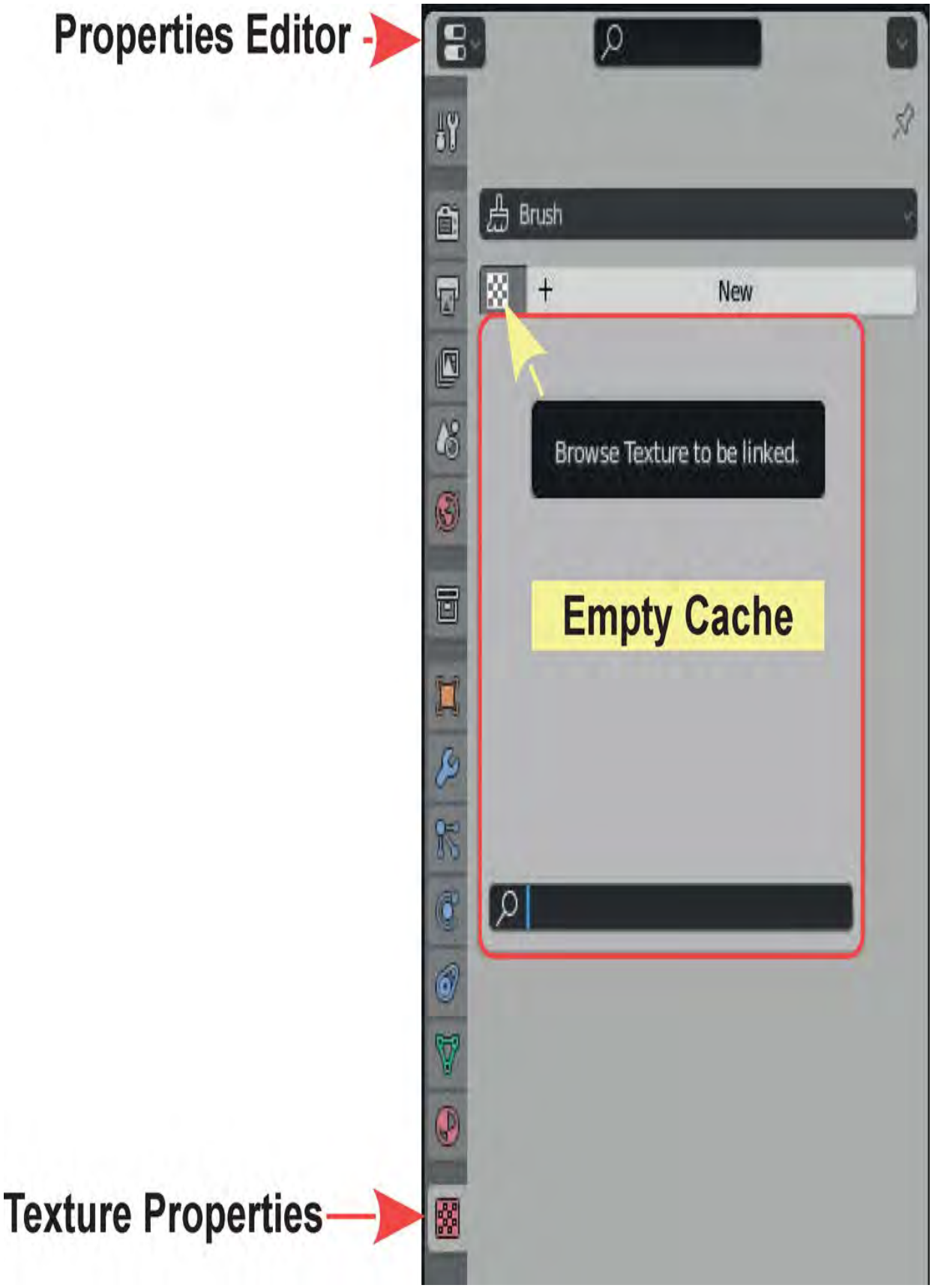

Properties Editor
Brush
New
Browse Texture to be linked.
Empty Cache
Texture Properties

Figure 17.4

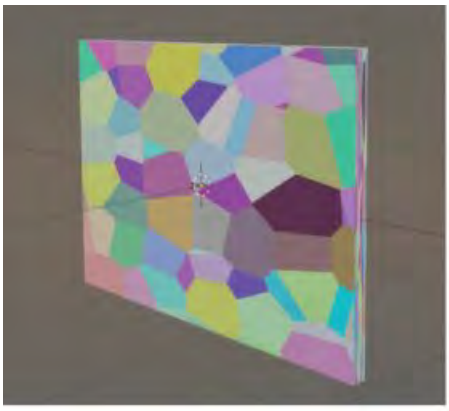
The **Brick Texture** is one of Blender's **Procedural Textures.** Experiment with different Textures on different Objects.

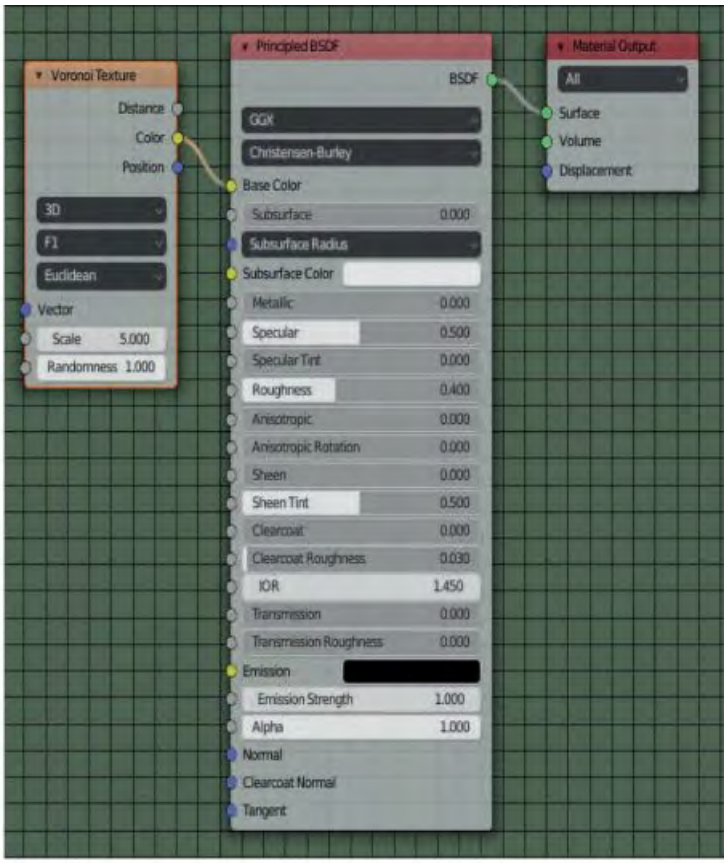
Texture Properties (Figure 17.4)

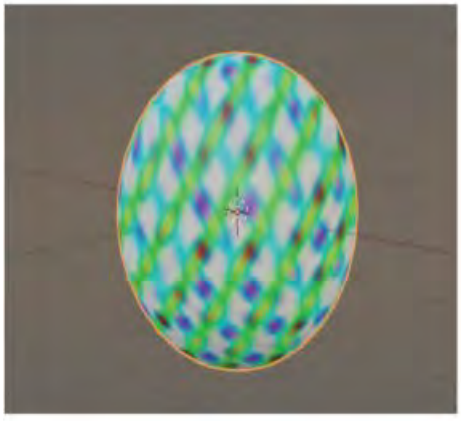
Note: By clicking the **Browse Texture to be linked button** in the **Properties Editor, Texture Properties** you display the **Texture Cache.** At this point the Cache is empty despite having applied a Texture (see section 17).

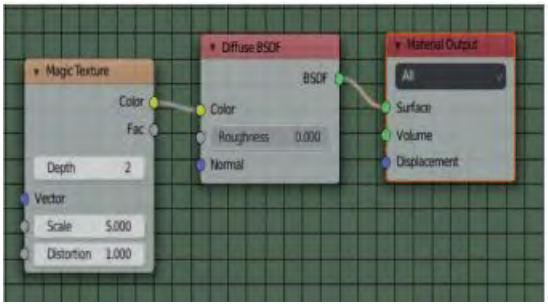
Procedural Texture Examples

Voronoi Texture




Voronoi Texture
Distance
Color
Position
3D
F1
Euclidean
Vector
Scale 5.000
Randomness 1.000
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius
Subsurface Color
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.400
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement




Magic Texture
Color
Fac
Depth 2
Vector
Scale 5.000
Distortion 1.000
Diffuse BSDF
BSDF
Color
Roughness 0.000
Normal
Material Output
All
Surface
Volume
Displacement

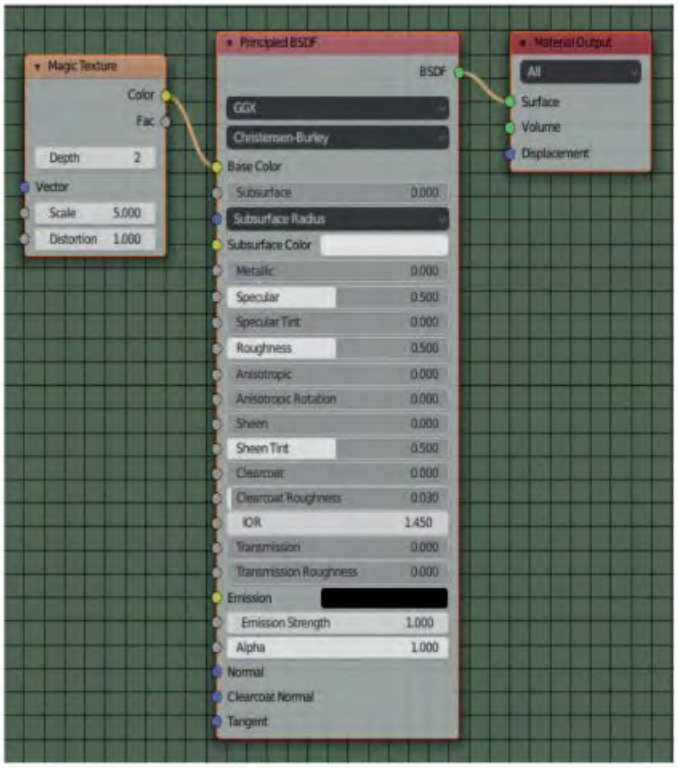

Magic Texture
Color
Fac
Depth 2
Vector
Scale 5.000
Distortion 1.000
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius
Subsurface Color
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.500
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement

Figure 17.6

In the examples the **Texture Nodes** have been connected to the **Base Color input socket** of the **Principled BSDF Node.** Since this is the only function of the BSDF Node being used the Principled BSDF Node may be replaced by the **Diffuse BSDF Node** producing the same result (Figure 17.6).

The **Procedural Textures** included in the default Blender file are but the tip of the iceberg. As you have seen, selecting one of the Textures from the menu connects a Texture Node to the Base Color input socket of the BSDF Node which outputs the result to the surface of the selected Object in the 3D Viewport Editor. Introducing other Nodes into the pipeline in the Shader Editor modifies the Texture output creating new Textures. The combination of Nodes and the output results are infinite and, therefore, impossible to demonstrate in a single publication. To discover the possibilities you will have to experiment and research the internet for examples. Having found a result you then record the findings for future use.

One excellent source is to be found on the Blender Artists website:

https://blenderartists.org/t/my-100-free-cycles-procedural-textures-blend-files-and-or-settings-included/1113432

An example of a Texture Node System is shown here in Figure 17.7.

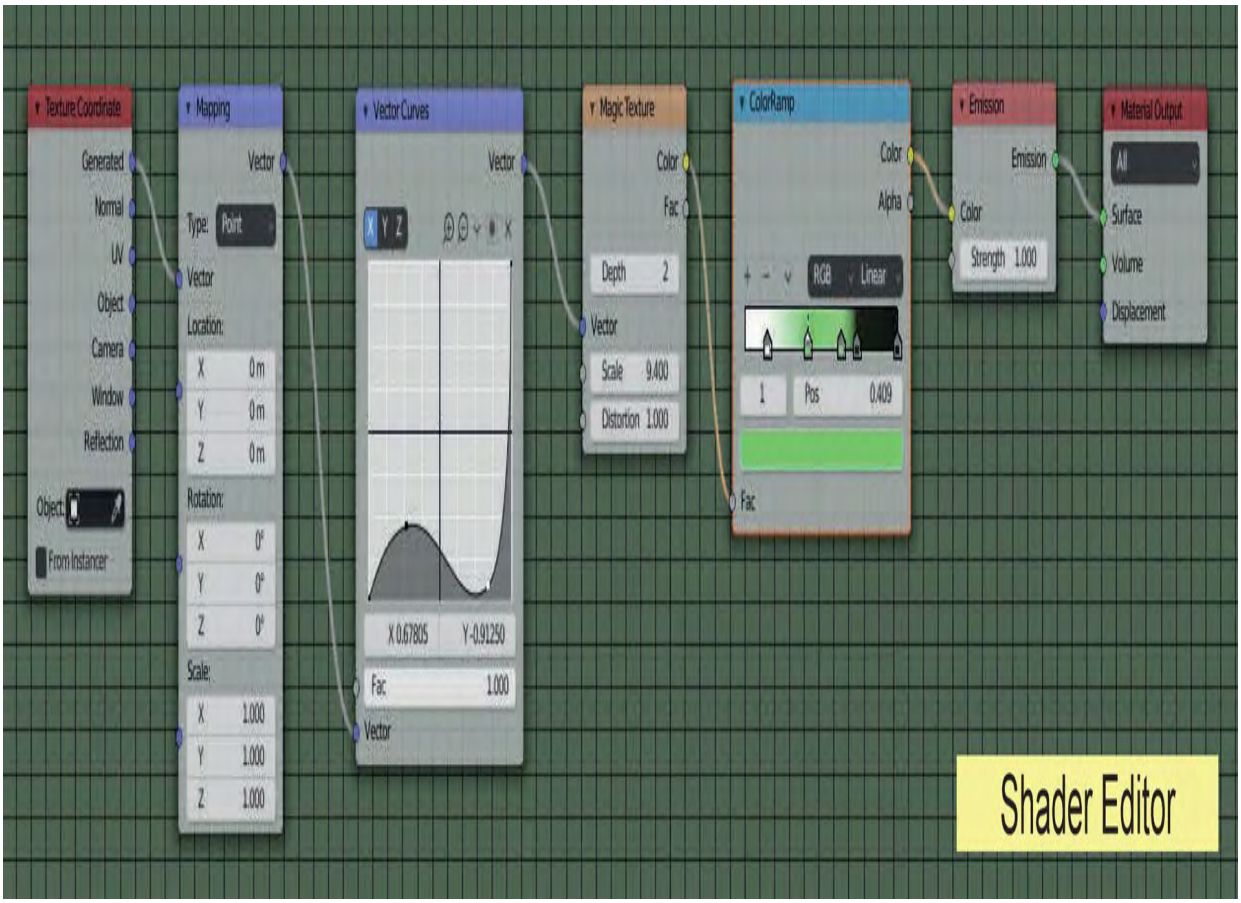


Figure 17.7

The Node Arrangement in Figure17.7 produces the pattern on the surface of a Plane Object (Figure 17.8).

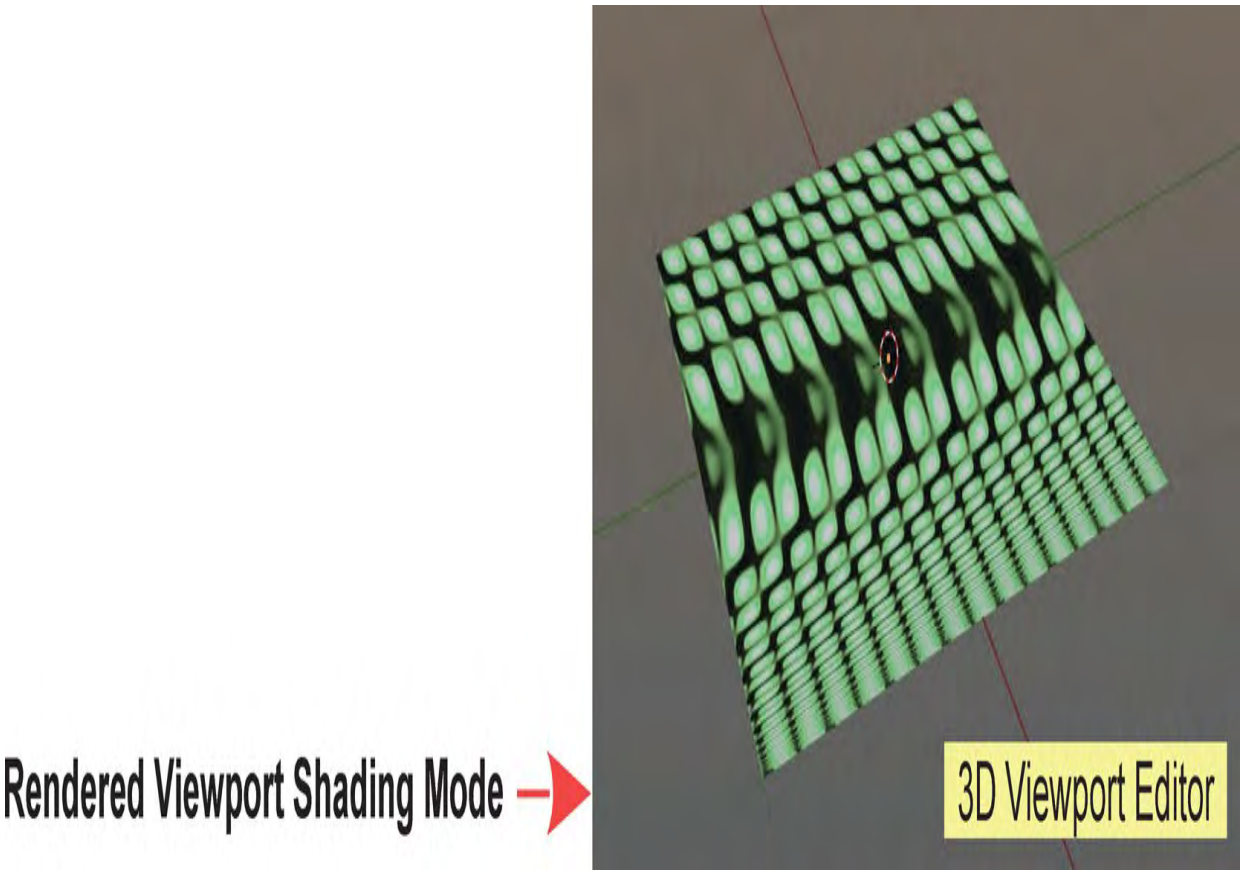


Figure 17.8

Note: The original Texture example has been designed for use with **Cycles Render** activated but also works with **Eevee Render.**

17.2 Image as a Texture

As previously mentioned an Image may be used as a Texture. The Image, such as JPEG, BMP or PNG, must be saved on your computer. One source of free Texture Images is: https://texture.ninja. These images are provided by Joost Vanhoutte; https://www.patreon.com/textureninja.

When downloading images to be used as Textures it is a good practice to save them to a dedicated folder which creates a Texture Library.

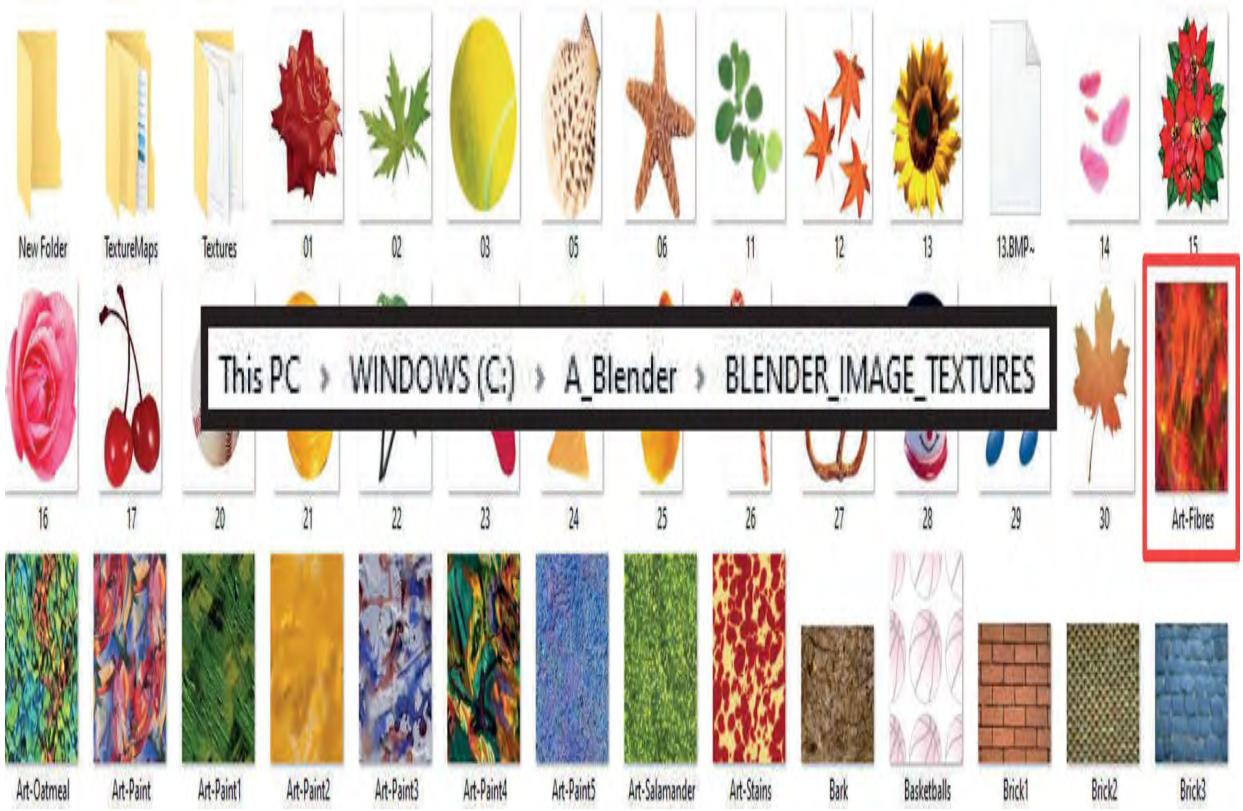


Figure 17.9

Applying an Image Textur

To use an Image as a Texture, apply a Material to an Object selected in the 3D Viewport Editor. Click the New Button, then in the **Properties Editor, Material Properties,** click the **Base Color button** and select, **Image Texture** from the menu. In the **Shader Editor** you will see an Image Texture Node connected to the Base Color input socket of the Principled BSDF Node (Figure 17.10).

In the Material Properties or in the Image Texture Node click on **Open,** navigate to the folder containing your Image and select it.

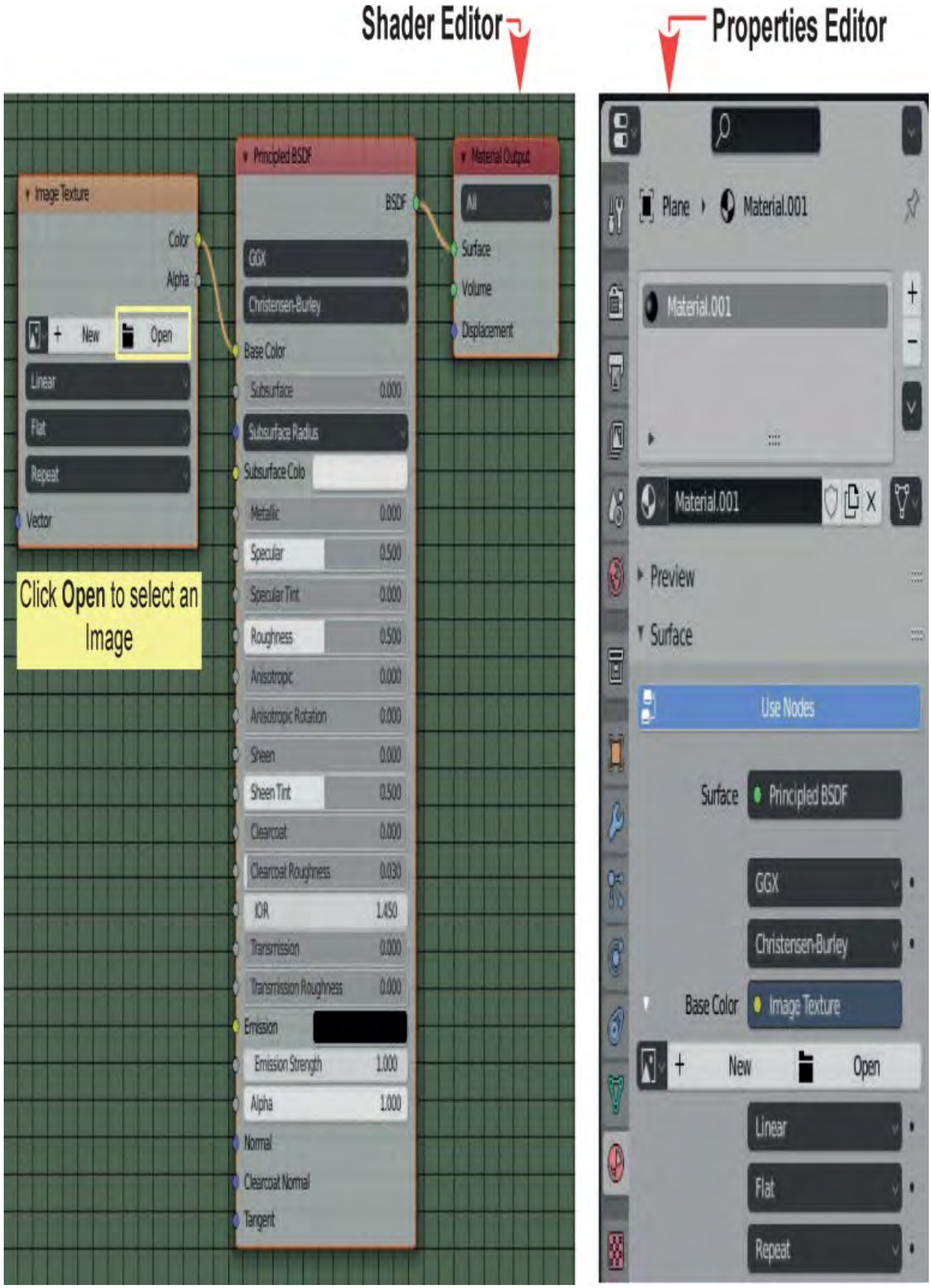

Shader Editor
Properties Editor
Image Texture
Color
Alpha
New
Open
Linear
Flat
Repeat
Vector
Click Open to select an Image
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius
Subsurface Colo
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.500
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement
Plane
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Image Texture
New
Open
Linear
Flat
Repeat

Figure 17.10

You will see the Image mapped to the surface of the selected Object in the **3D Viewport Editor** (in Material Preview or Rendered Viewport Shading Mode — Figure 17.11).

Image Texture named **Art-Fibres** (Figure 17.9) selected, **Mapped** to a Plane Object in the 3D Viewport Editor.

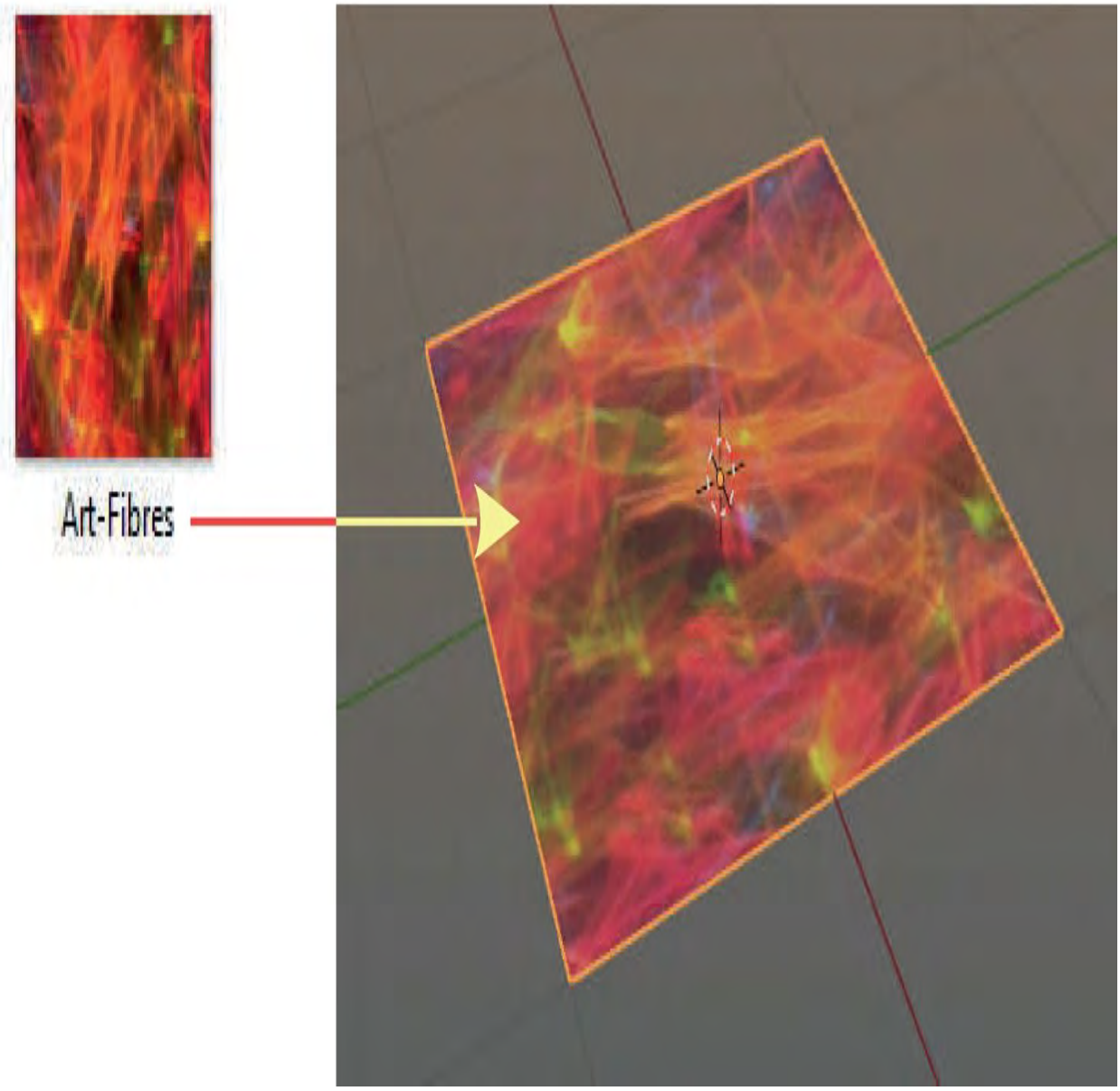


Figure 17.11

Mapping: How the texture is configured on the surface of an Object.

17.3 Texture Mapping

Mapping is best demonstrated with an Object a little more complicated than a flat Plain Object which for all tense and purposes has a single surface if you disregard its reverse side. The default Cube Object will do nicely. A Cube has six sides, therefore, six surfaces.

The default Cube has a Material pre-applied (gray).

For the demonstration a JPEG Image of a landscape will be used which provides a variation in detail instead of an indistinguishable pattern of color.

Figure 17.12

Mapping the Image to the surface of the Cube means that the Image or parts of the Image will be transferred to the six sides of the Cube. To do this the Cube will be laid out flat as if it were a cardboard box undone at the seams.

To see the unwrapping effect have the default Cube Object in the 3D Viewport Editor selected in **Edit Mode.** Divide the 3D Viewport Editor in two and change the left hand part to the **UV Editor.**

The default Cube is automatically laid flat (Unwrapped) in the **UV Editor** when the Object is in **Edit Mode.** The unwrapping fits inside the confines of what will be the outline of your Image when the Image is loaded in the UV Editor (Figure 17.13 over).

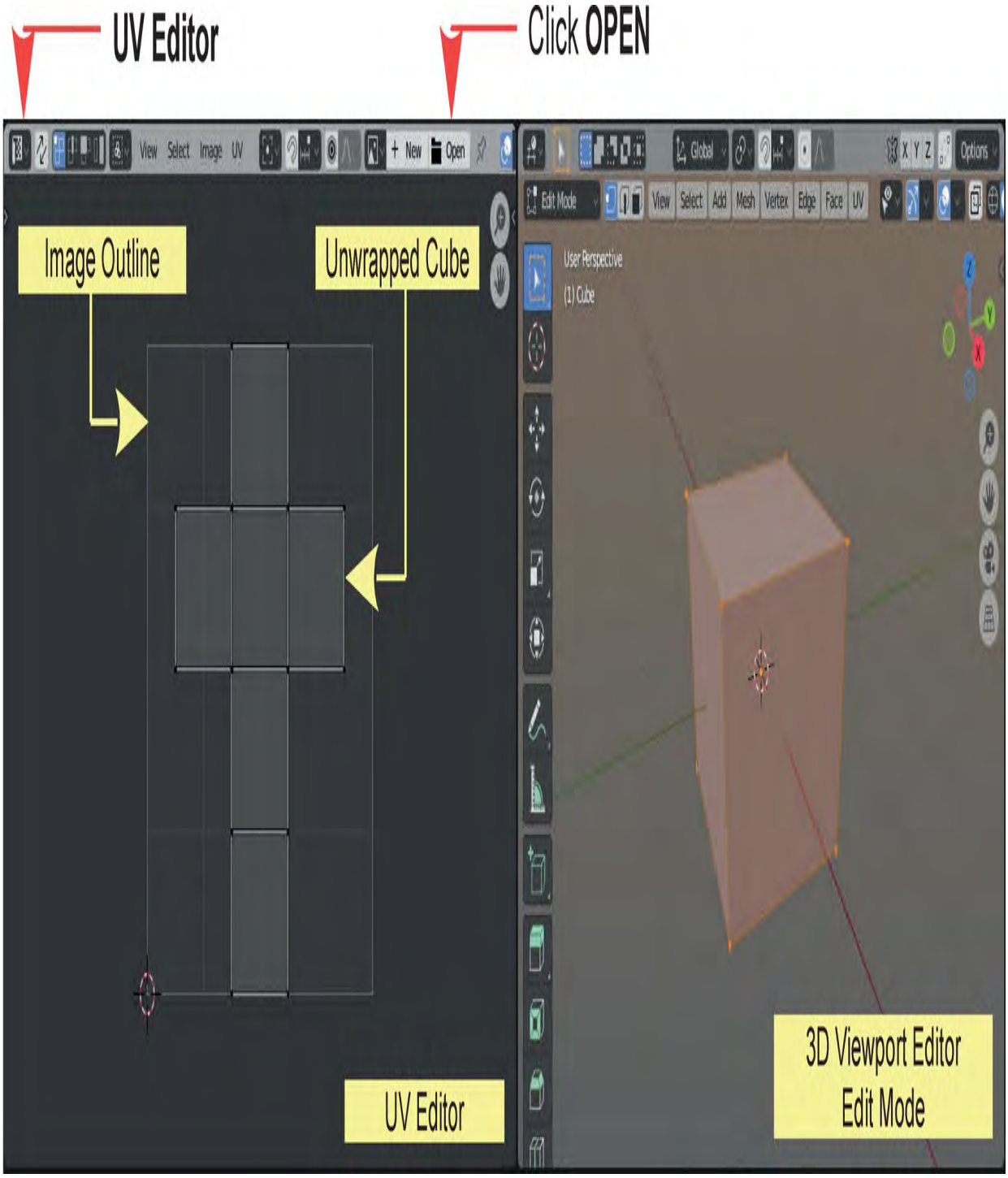


Figure 17.13

At this point the Image has not been loaded in the UV Editor nor has it been assigned as an Image Texture for the Cube Object. You will not see the Image applied to the surface of the Cube.

In the **UV Editor,** click on **Open** in the Header (Figure 17.13), navigate to your image and select it. This enters the Image in the UV Editor where the original Image confines change to fit the Image selected and the unwrapped Cube deforms to fit the new confine Figure 17.14).

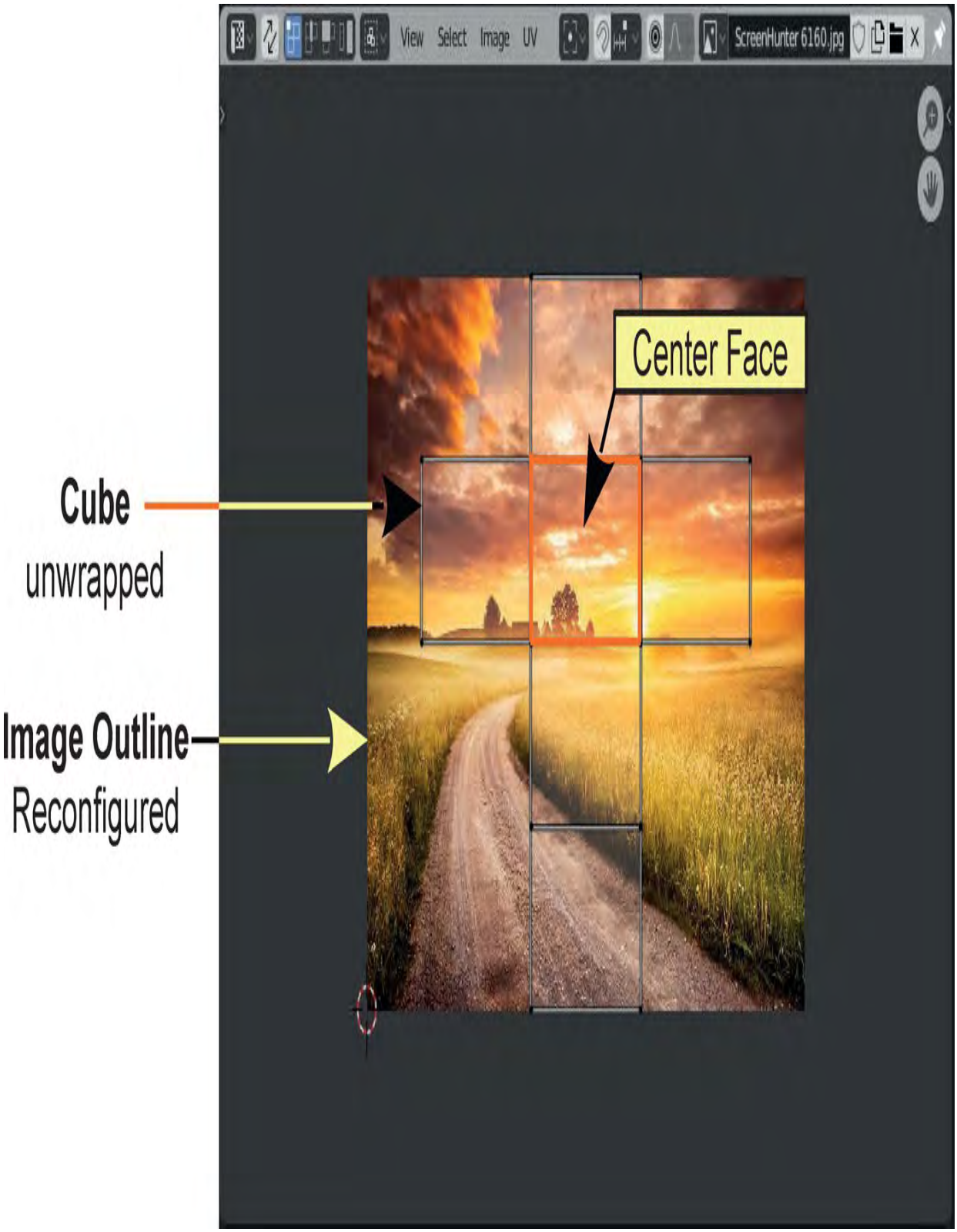

View
Select
Image
UV
ScreenHunter 6160.jpg
Center Face
Cube
unwrapped
Image Outline
Reconfigured

Figure 17.14

Note: At this point the Image has not been assigned as an Image Texture to the Cube Object and, therefore, **does not** display on the surface of the Cube in the 3D Viewport Editor.

You assign the Image as a Texture, in the **Properties Editor, Material Properties,** click the **Base Color button,** select **Image Texture** and open the Image in the **Image Texture Node** (Figure 17.11).

With the Image assigned as a Texture it displays in the 3D Viewport Editor, Mapped to the surface of the Cube Object. Note the center face of the Cube in the UV Editor and the equivalent face of the Cube in the 3D Viewport Editor and the orientation of the Mapped Image.

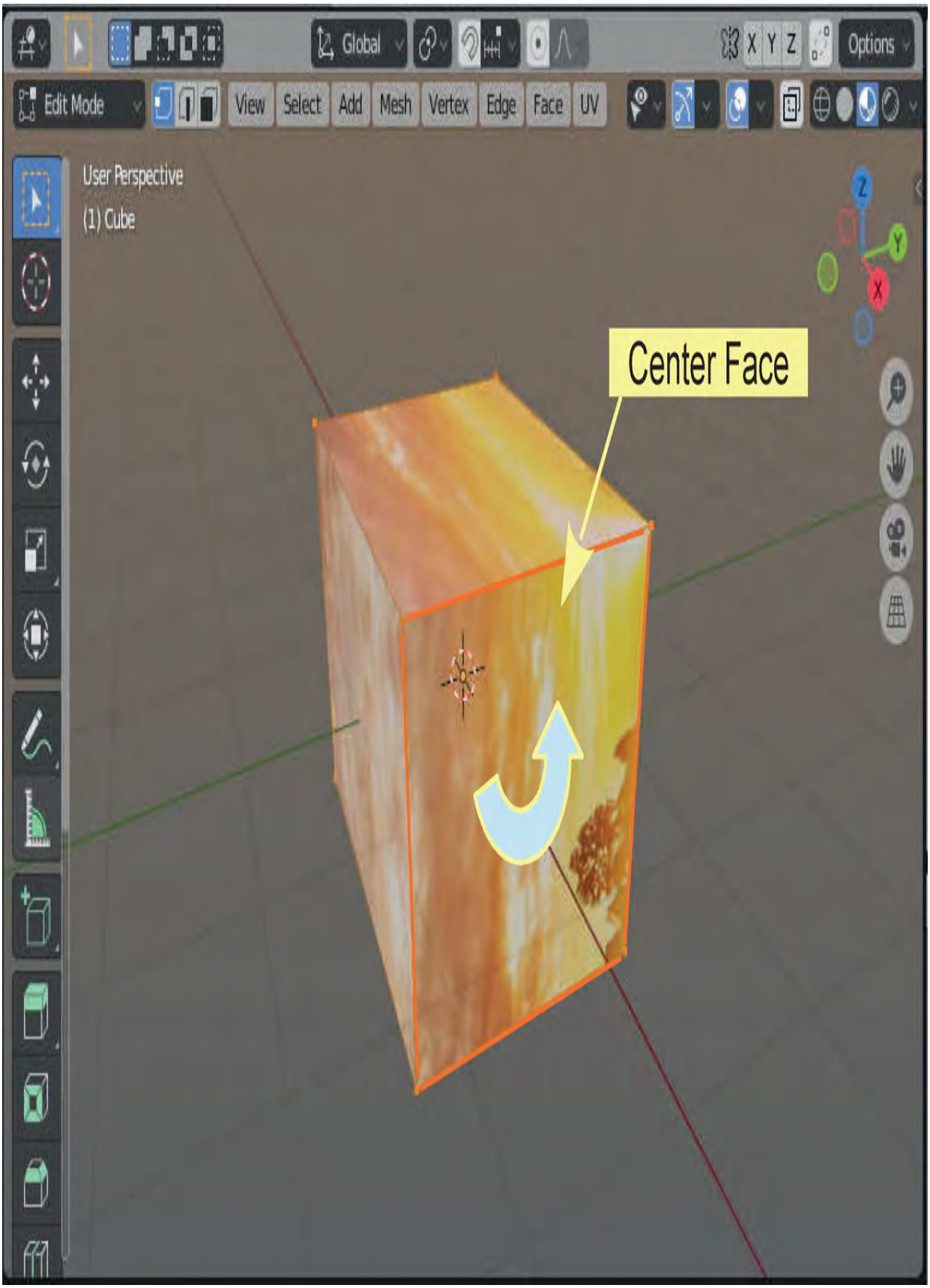

Global
X Y Z
Options
Edit Mode
View
Select
Add
Mesh
Vertex
Edge
Face
UV
User Perspective
(1) Cube
Center Face

Figure 17.15

By clicking a point or points (vertices) in the **UV Editor,** pressing the **G Key** and moving the points you may change which portion of the Image is mapped to a Face on the Cube. In Figure 7.16 the Cube has been rotated about the X Axis in Object Mode and returned to Edit Mode.

Figure 17.16

17.4 Unwrapping Options

The Unwrapping method has been demonstrated using the default automatic **Unwrap.** Different options are found in the **UV Editor Header, UV Button** (Figure 17.17).

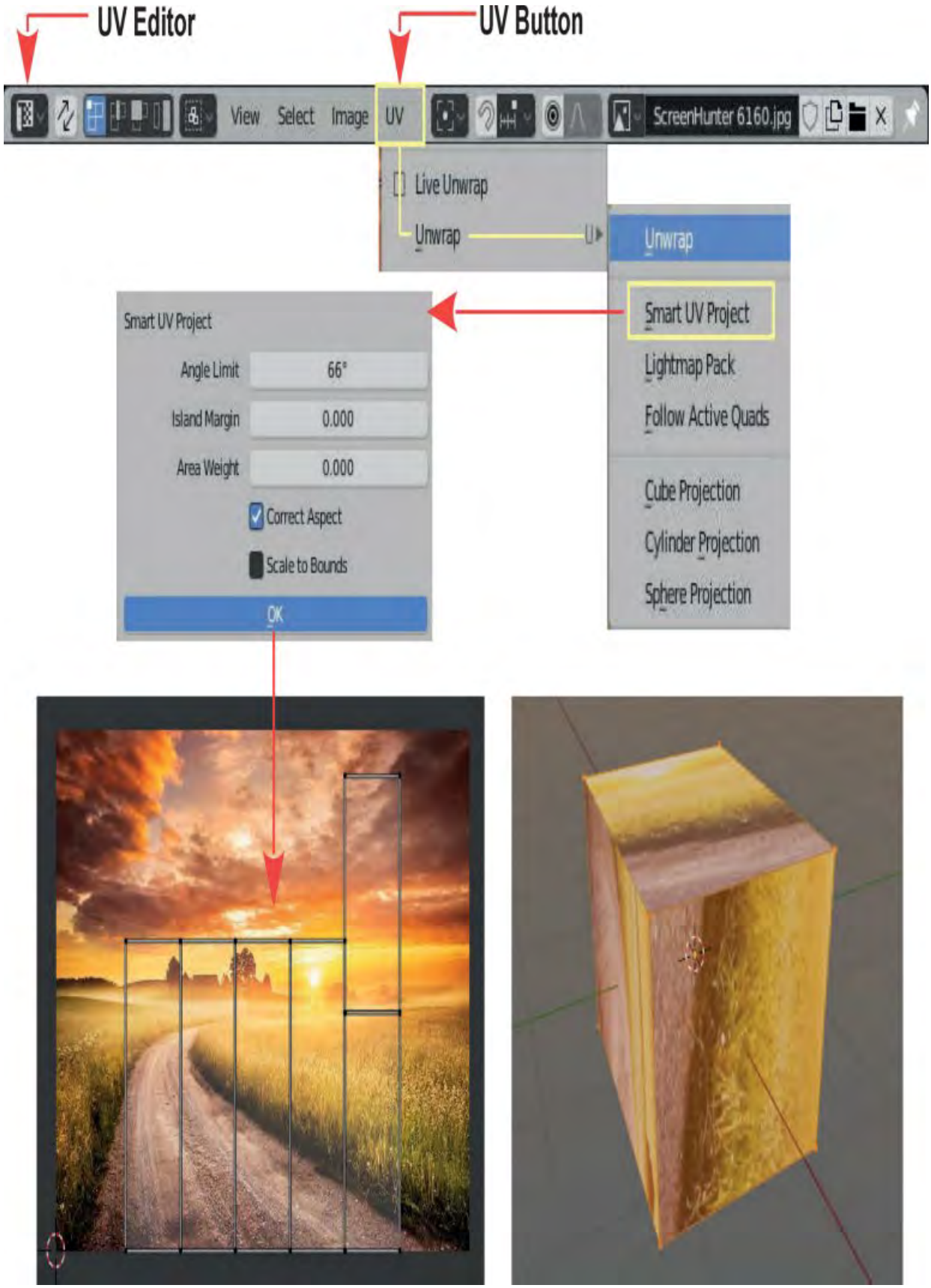

UV Editor
UV Button
View
Select
Image
UV
ScreenHunter 6160.jpg
Live Unwrap
Unwrap
Unwrap
Smart UV Project
Lightmap Pack
Follow Active Quads
Cube Projection
Cylinder Projection
Sphere Projection
Smart UV Project
Angle Limit
66°
Island Margin
0.000
Area Weight
0.000
Correct Aspect
Scale to Bounds
OK

Figure 17.17

17.5 PBR Textures

PBR Textures (Physically Based Rendering) are a Texture workflow to simulate the lumps and bums on a surface which react with light, as do real life Materials. PBR Textures in practice are JPEG or PNG Images configured to represent dark and light areas on a surface which include highlights and shadows. A combination of PBR Texture Images are employed to create a Textured surface. The Images are entered into Texture Nodes and configured in the Shader Editor.

PBR Texture Images may be generated for specific applications but are generally downloaded from Texture Repositories on the internet similar to downloading blocks of computer code.

To demonstrate the basics of PBR Texturing a **Texture Pack** (series of Images) has been downloaded from: https://www.virtualtweaks.com/ **Free PBR Shader — terraCottaBB.zip**

When unzipped **terraCottaBB.zip** creates seven (7) Images and an Information file giving permission to use and explaining that the Images are low resolution and intended for practice.

In demonstrating the use of the PBR Textures not all images will be used. When downloading any Texture Pack the first hurdle in using the Textures is understanding which Images are applicable.

A detailed explanation will not be attempted at this stage, instead a practical example of implementing the Images will be presented.

Download terraCottaBB.zip and unzip the file to a folder. The folder will contain:

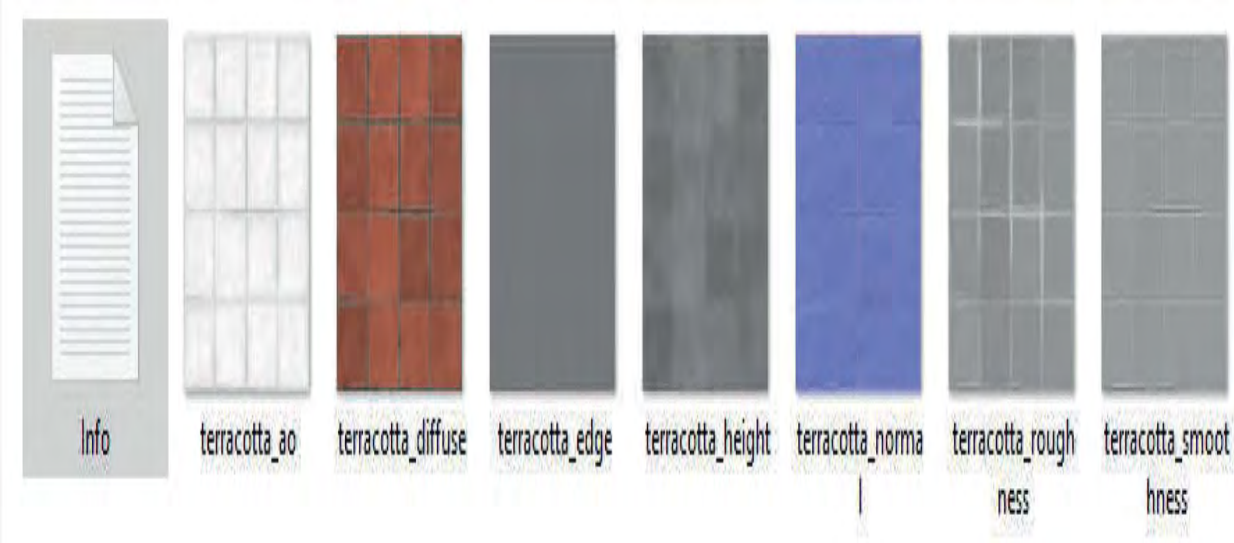

Info
terracotta_ao
terracotta_diffuse
terracotta_edge
terracotta_height
terracotta_normal
terracotta_roughness
terracotta_smoothness

Figure 17.18

In Blender change the default Layout Workspace to the **Shading Workspace** (select in the Header).

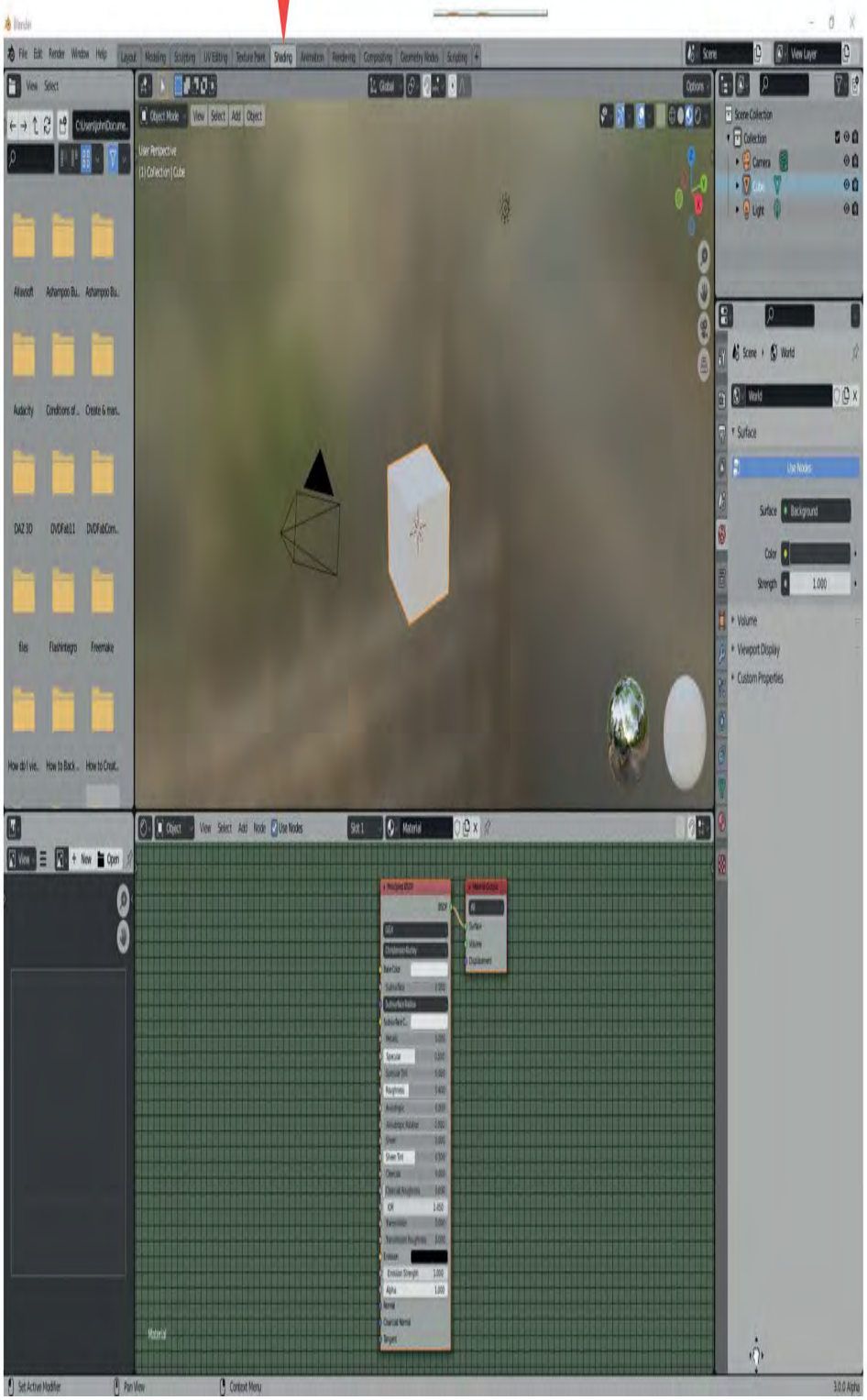


Figure 17.19

In the **File Browser Editor** (upper LHS) navigate to and open the folder containing the **PBR Texture Files.** By default the folder opens in **Thumbnail display mode** where the names of the files are obscured. By changing to one of the **List display modes** you see details of each file.

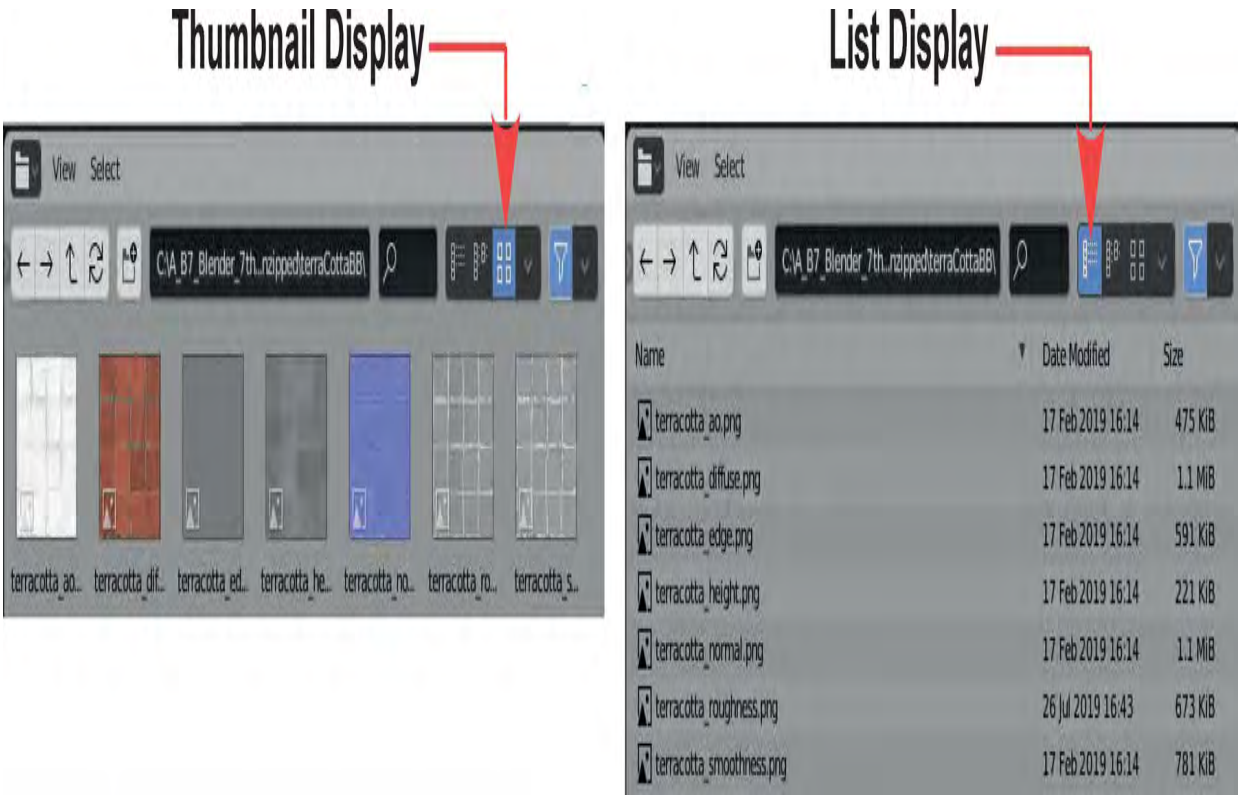


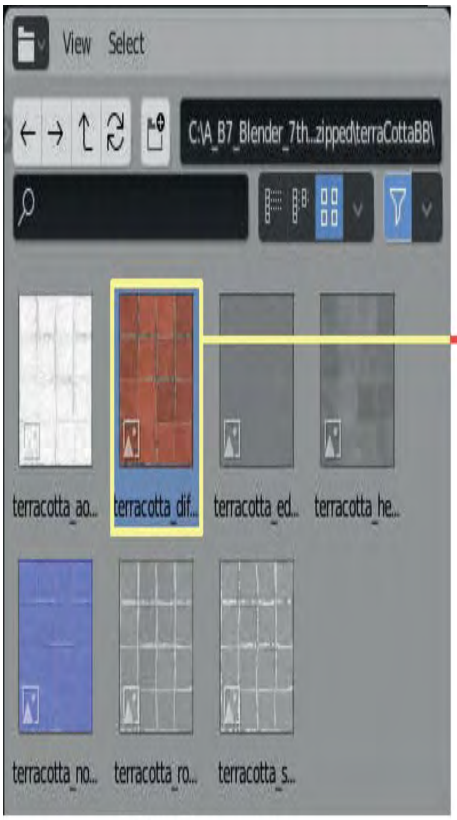
Figure 17.20

To demonstrate the application of PBR Textures, delete the default Cube from the 3D Viewport Editor and replace it with a Plane Object. **Add a Material.** Remember; the default Cube has a Material pre-applied. A new Object added to the Scene does not have a Material until you press the New Button in the Properties Editor, Material Properties.

With a Material applied and Use Nodes active you see the Principled BSDF Node connected to the Material Output Node in the Shader Editor.

The PBR Texture Thumbnails may be selected in the File Browser and dragged into the Shader Editor automatically entering an **Image Texture Node** with the PBR Texture Image entered in the Node.

Part A. Select and drag the **terracotta_diffuse.png** PBR Image into the Shader Editor and connect the Color output socket of the Image Texture Node to the Base Color input socket of the Principled BSDF Node.



File Browser

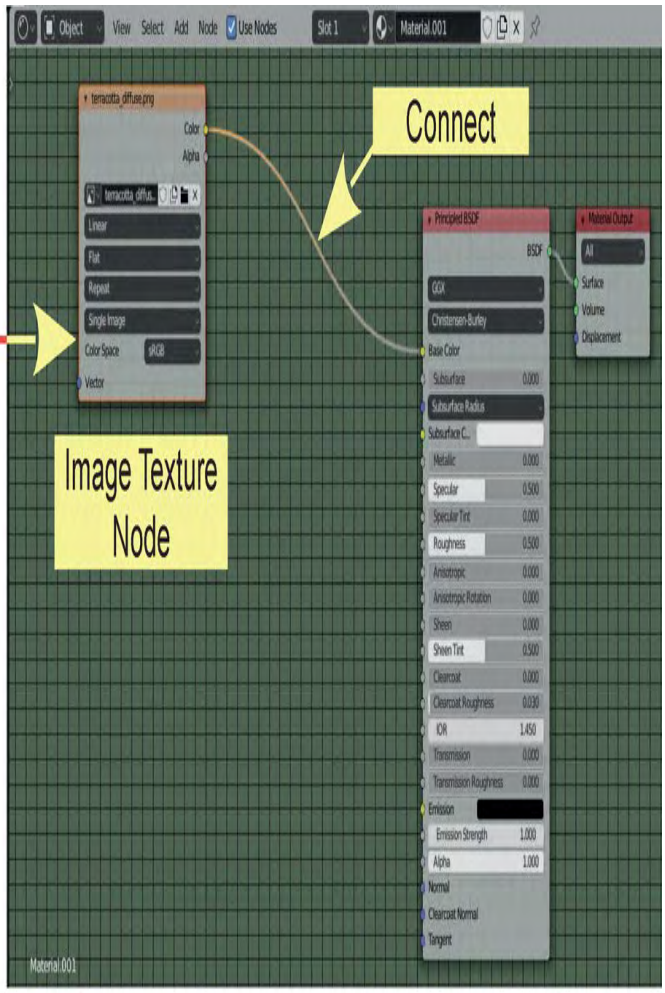


Figure 17.21

With the Nodes connected and with the 3D Viewport Editor in Material Preview or Rendered Viewport Shading Mode you see the Terracotta Image Mapped to the surface of the Plane. Note: The Plane is a flat surface without any detail of surface imperfections (Lumps or Bumps).

The PBR Texture Images are entered into the Shader Editor and connected to the Principled BSDF Node in the following sequence **(Note:** Not all the PBR Texture Images in the texture Pack are used).

A.	terracotta_diffuse. png	provides the base color information for the Texture.
B.	terracotta_normal. png	simulates bumpiness on the base color.
C.	terracotta_height.png	allows adjustment of the bumpiness.
D.	terracotta_roughness.png	provides fine imperfections to the surface for realism.

Apart from **A. terracotta_diffuse.png** which is connected directly to the Diffuse BSDF Node the other Images/Nodes are connected and employed via intermediary Nodes.

Part B. Select and drag the **terracotta_normal.png** PBR Image into the Shader Editor and connect as shown. In the terracotta_normal.png Node change **Color Space** to **Non-Color** and change the values in the **Normal map Node** as shown.

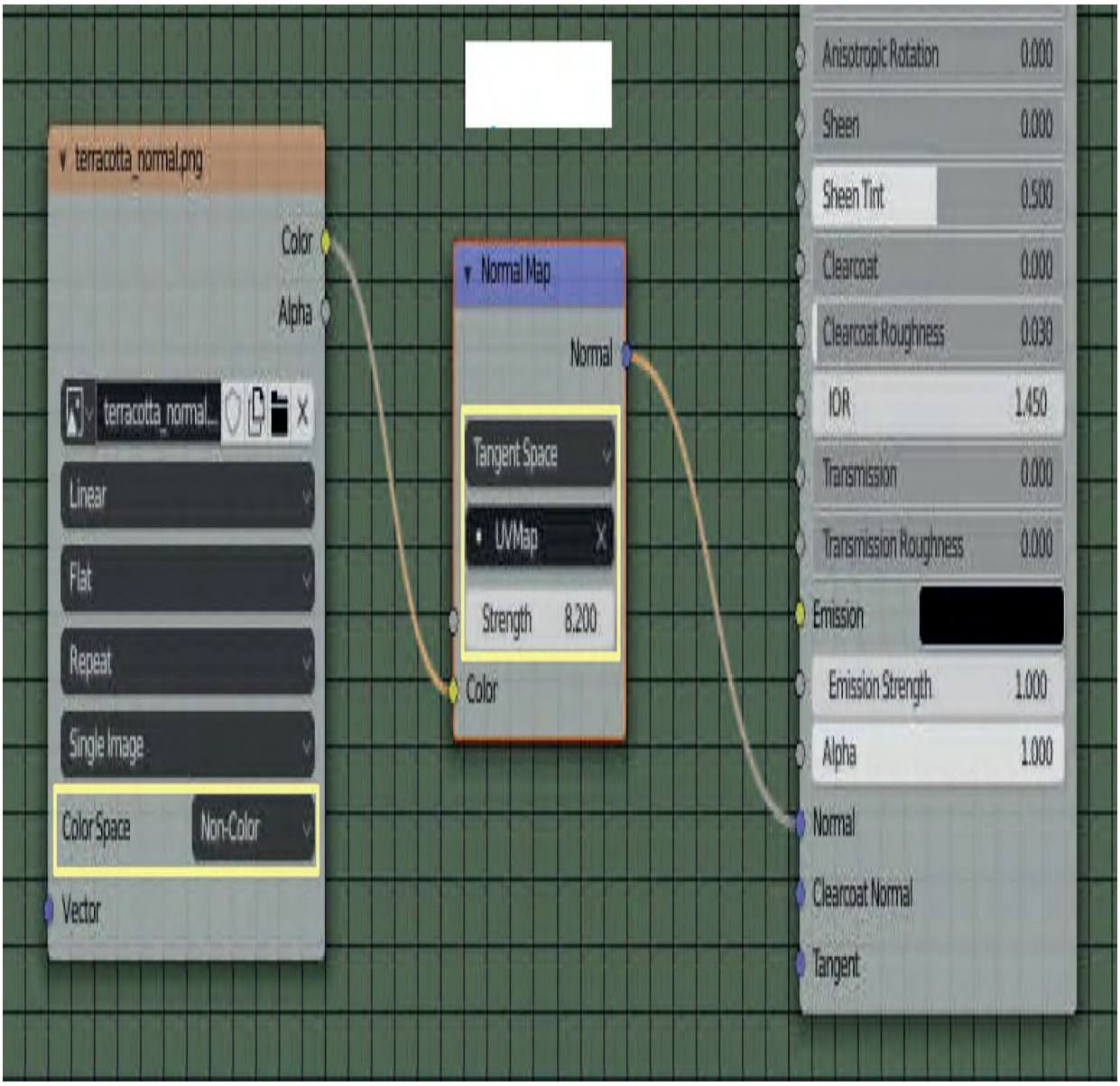


Figure 17.22

With the Nodes connected you will see how light reacts with the surface. Have the 3D Viewport Editor in rendered Viewport Shading Mode and locate the default Point Lamp down close to the Plane. In Figure17.24 the Point Lamp Power is set at 708.2 W.

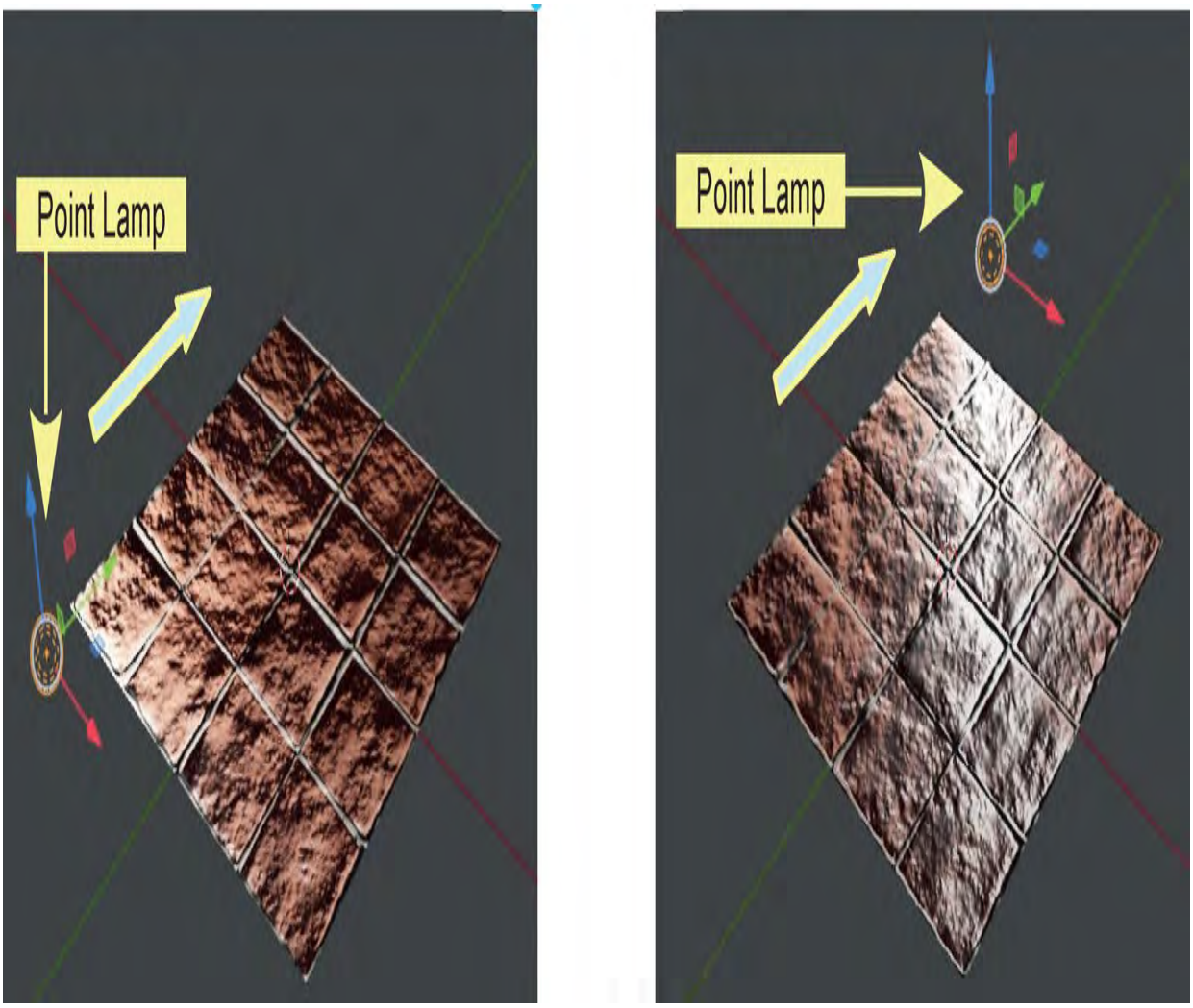


Figure 17.23

Part C The surface of the Plane may be further enhanced by the introduction of an **Image Texture Node** with **terrracotta_height.png** inserted. The Node is connected via a **Converter -RGB to BW Node** and a **Vector-Bum**.

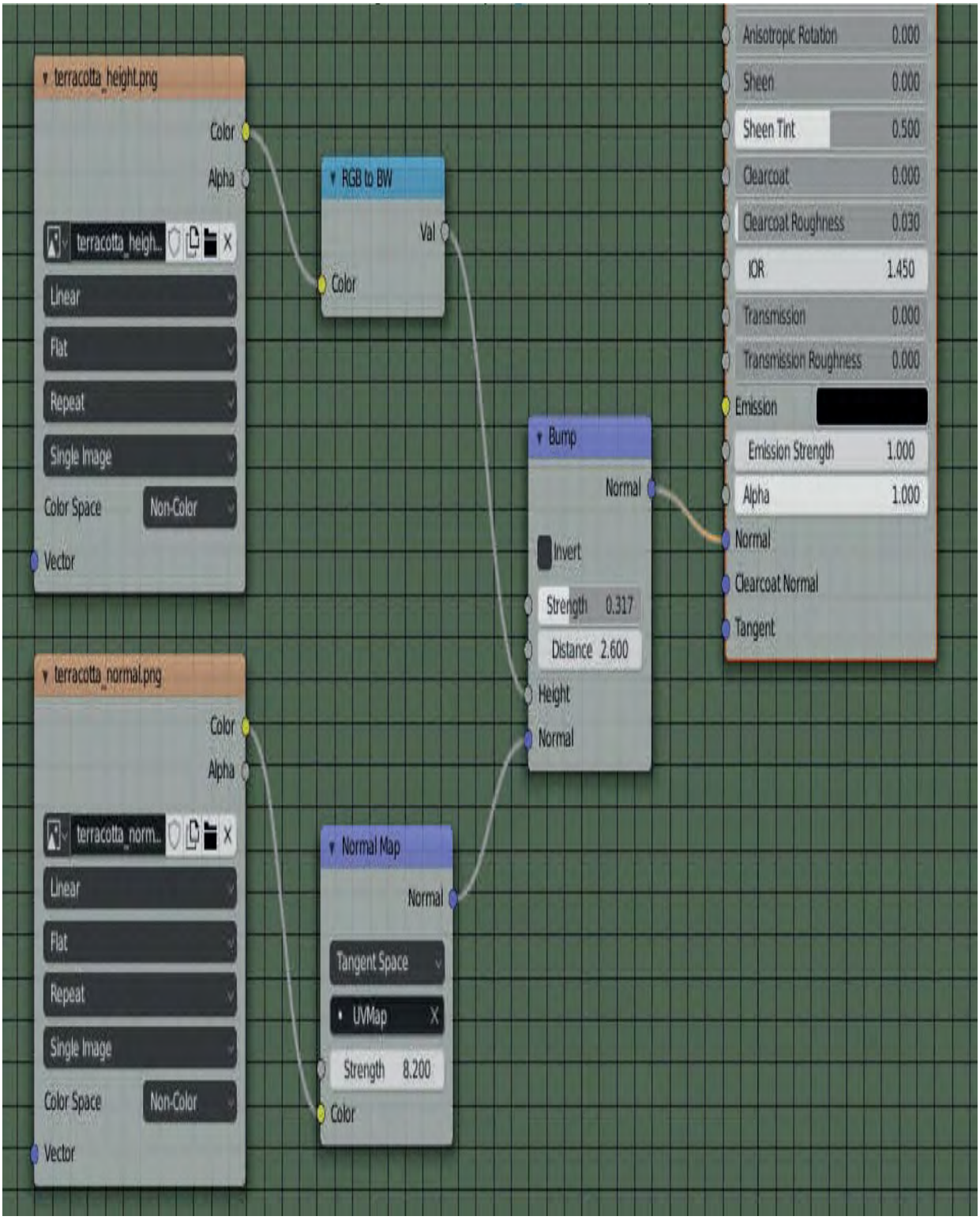

terracotta_height.png
Color
Alpha
terracotta_heigh...
Linear
Flat
Repeat
Single Image
Color Space
Non-Color
Vector
RGB to BW
Val
Color
Bump
Normal
Invert
Strength 0.317
Distance 2.600
Height
Normal
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
terracotta_normal.png
Color
Alpha
terracotta_norm...
Linear
Flat
Repeat
Single Image
Color Space
Non-Color
Vector
Normal Map
Normal
Tangent Space
UVMap
Strength 8.200
Color

Figure 17.24

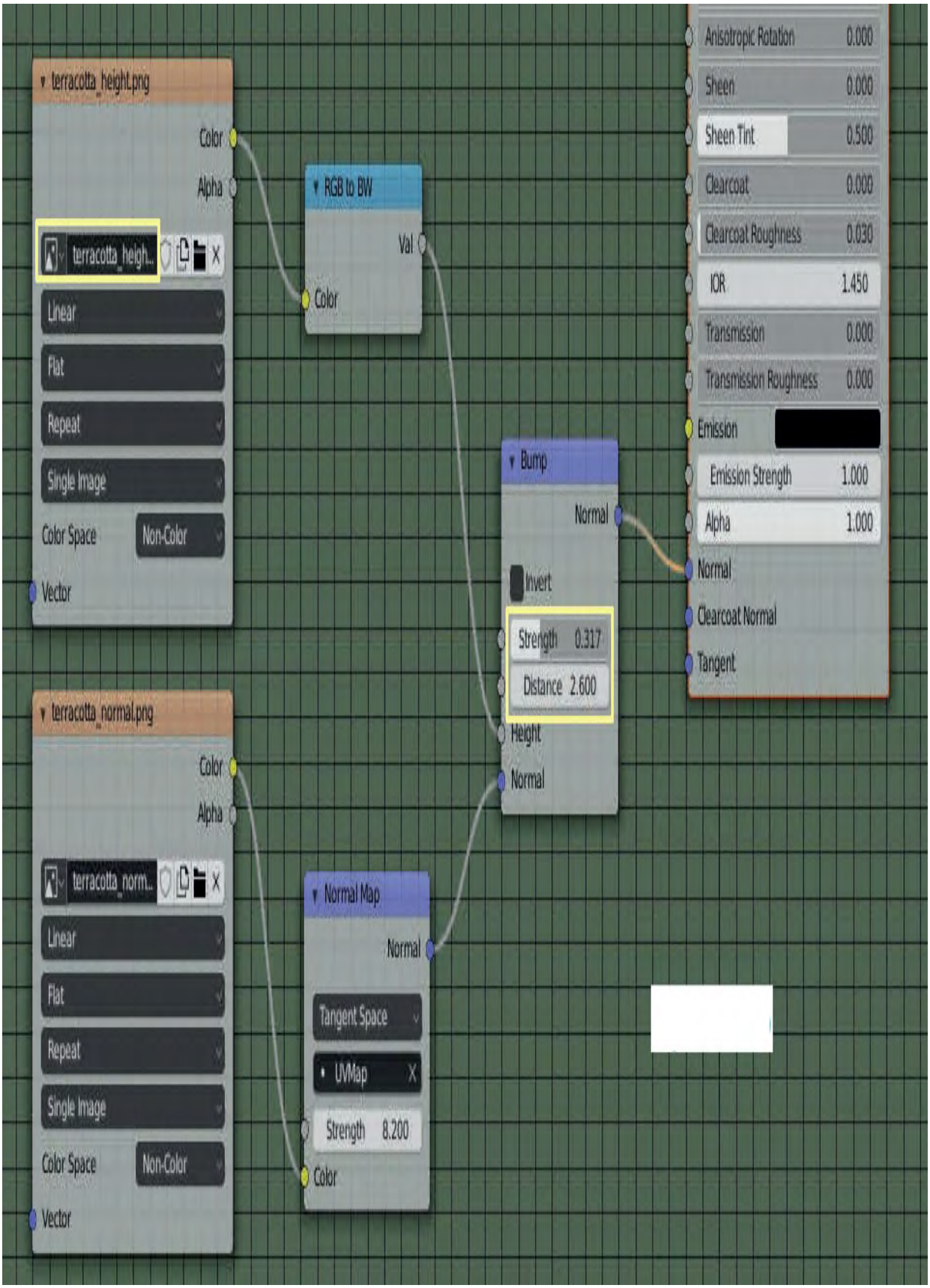

terracotta_height.png
Color
Alpha
terracotta_heigh...
Linear
Flat
Repeat
Single Image
Color Space
Non-Color
Vector
RGB to BW
Val
Color
Bump
Normal
Invert
Strength 0.317
Distance 2.600
Height
Normal
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
terracotta_normal.png
Color
Alpha
terracotta_norm...
Linear
Flat
Repeat
Single Image
Color Space
Non-Color
Vector
Normal Map
Normal
Tangent Space
UVMap
Strength 8.200
Color

Figure 17.25

The affect is controlled by adjusting the Strength and Distance values in the Bump Map Node.

By placing the 3D Viewport Editor in Front Orthographic View you will see that the Plane remains a flat surface (Figure 17.27).

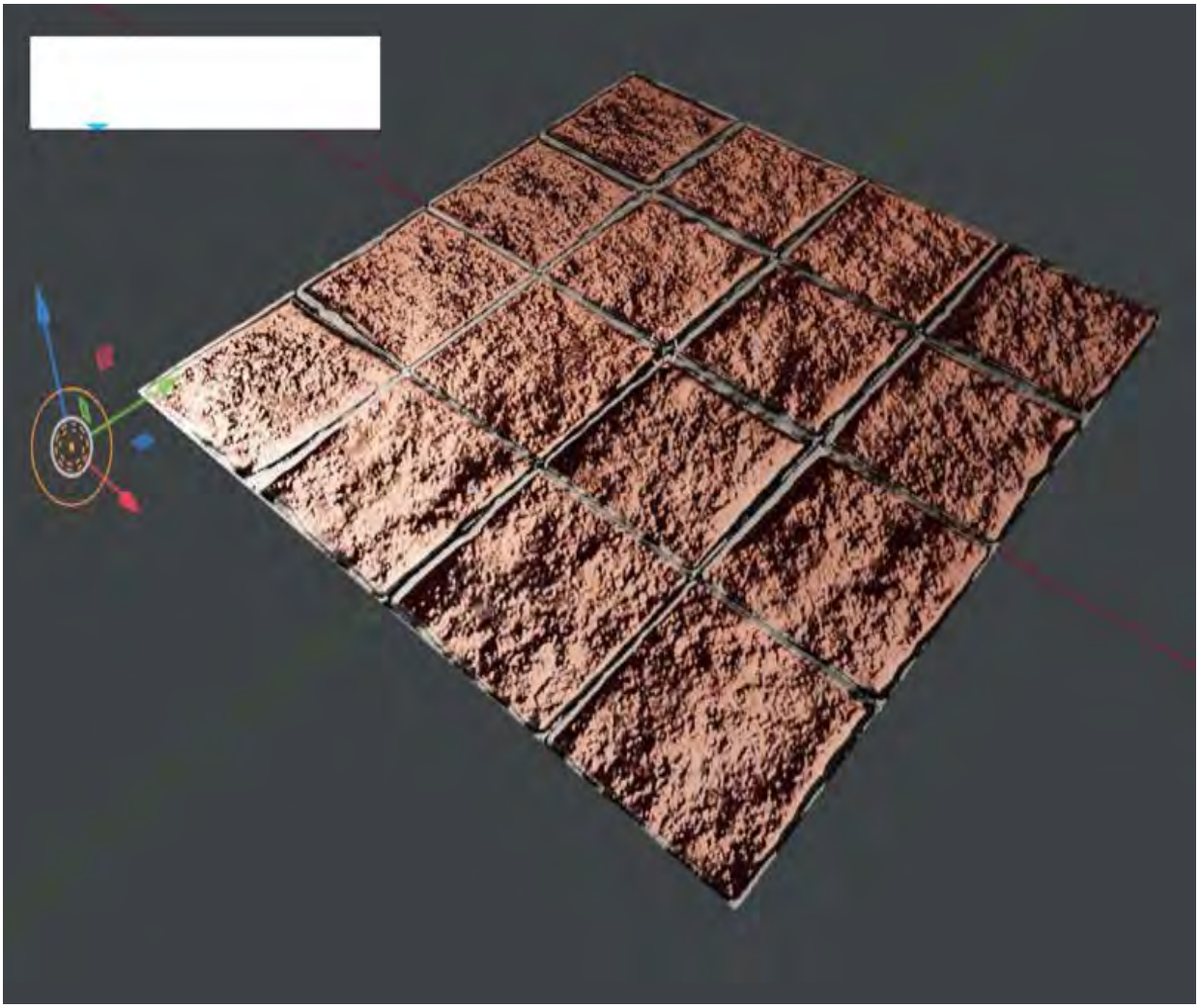


Figure 17.26

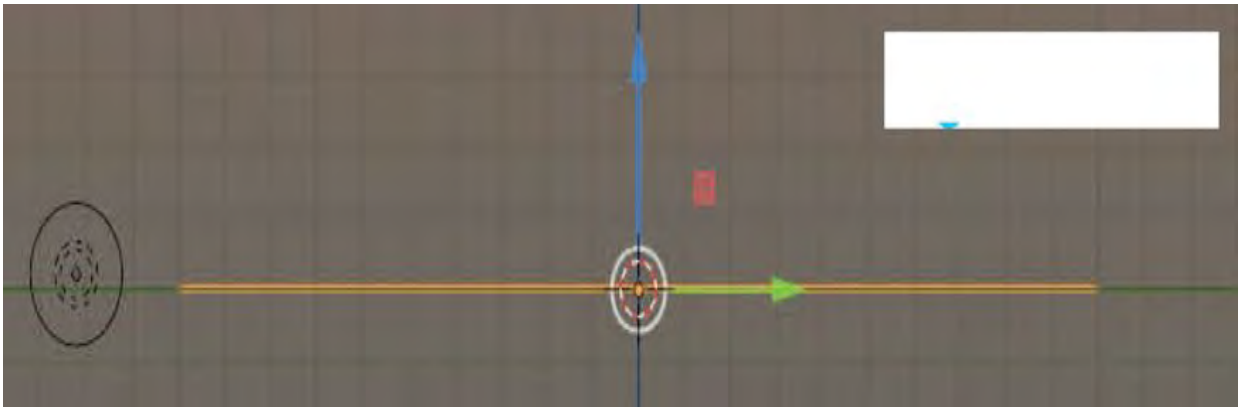


Figure 17.27

Part D. A final adjustment is obtained by introducing an Image Texture **terracotta_Roughness.png,** inserted and connected to the Roughness input socket of the Principled BSDF Node via a **Color Ramp Node.** By moving the Color Ramp Stops (Reference Chapter 16 — 16.22) fine tuning of the surface imperfections are made.

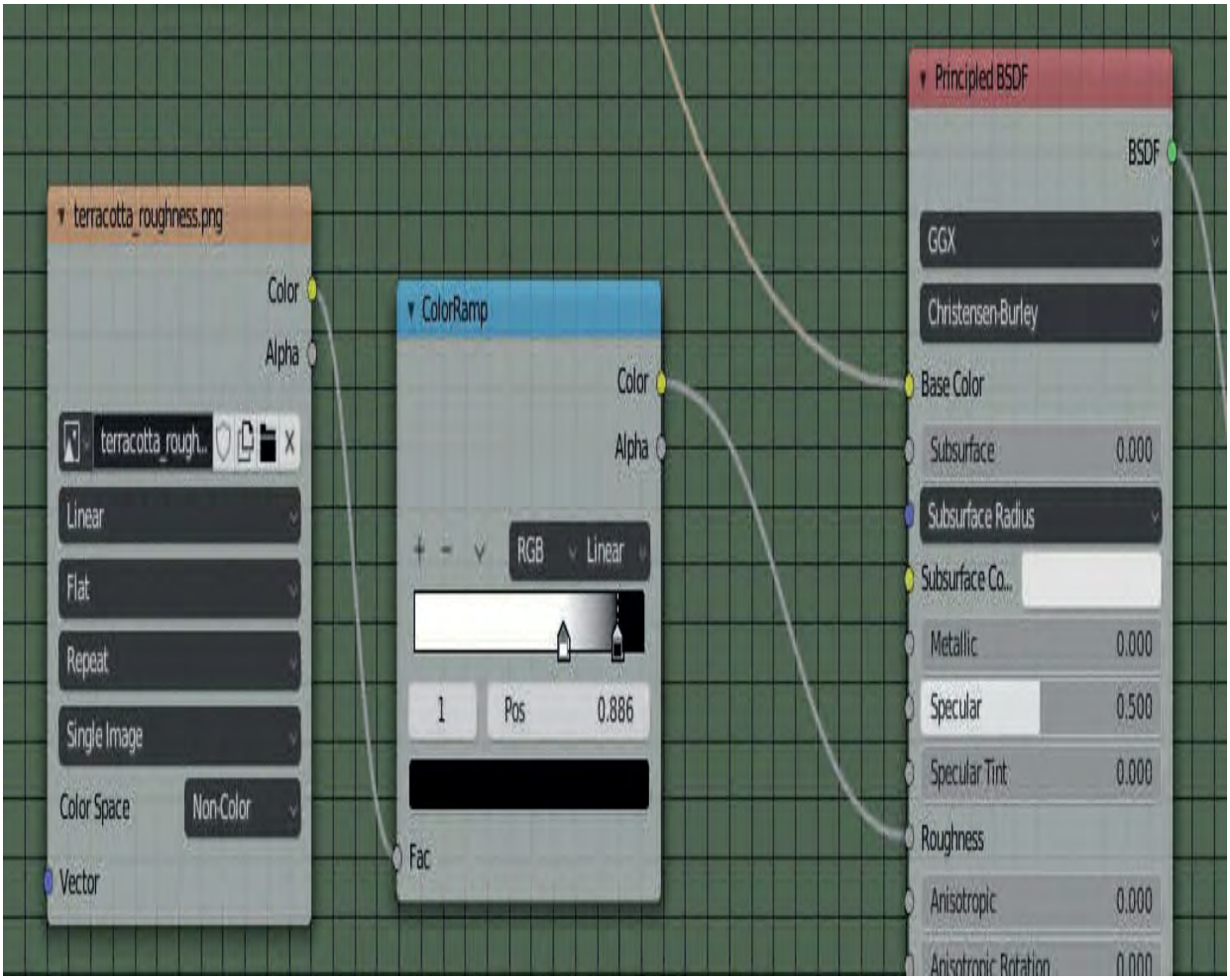


Figure 17.28

PBR Texturing Node Configuration

The completed PBR Texture Node configuration is shown in Figure 17.29.

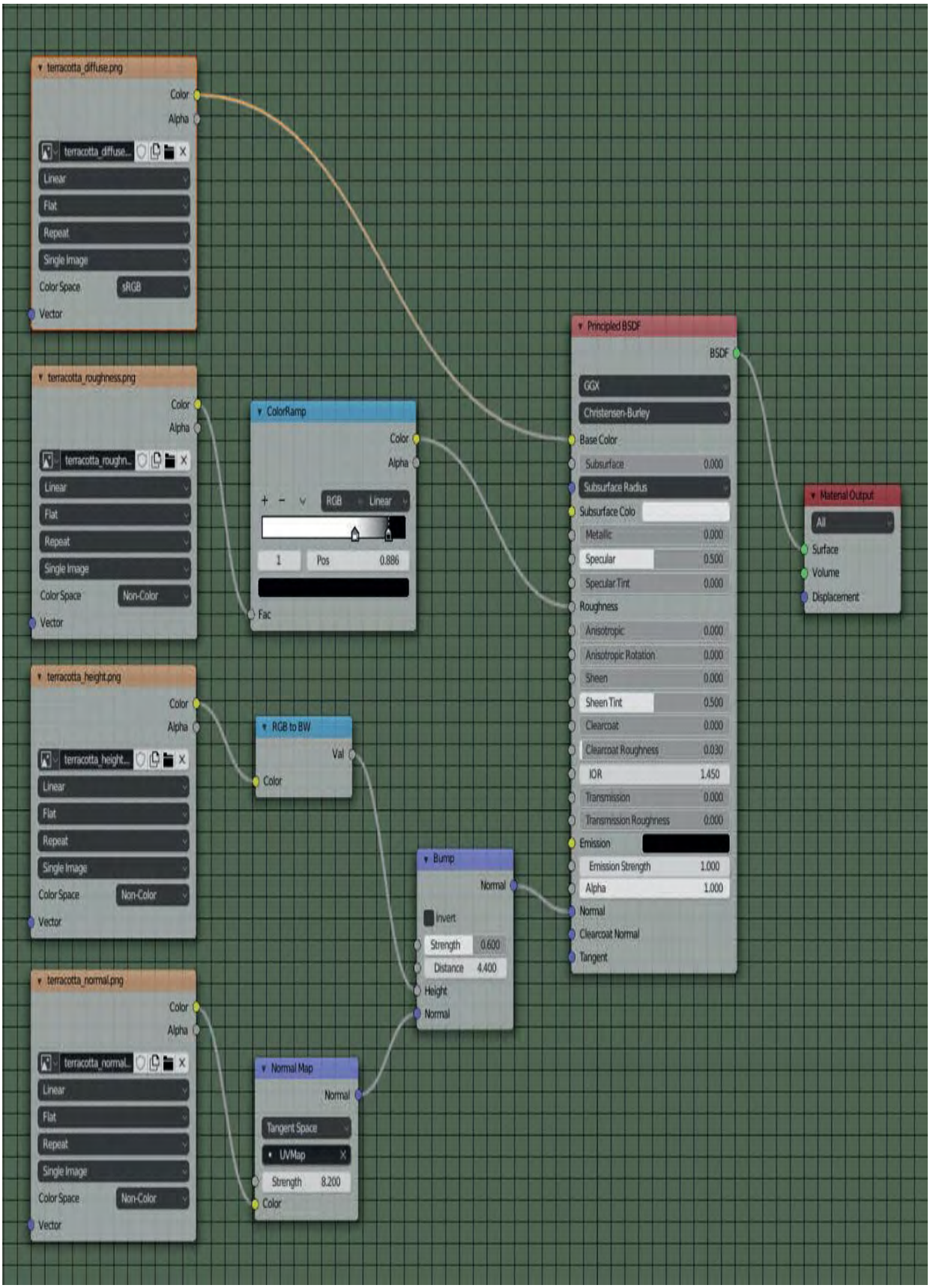

terracotta_diffuse.png
Color
Alpha
terracotta_diffuse...
Linear
Flat
Repeat
Single Image
Color Space
sRGB
Vector
terracotta_roughness.png
Color
Alpha
terracotta_roughn...
Linear
Flat
Repeat
Single Image
Color Space
Non-Color
Vector
ColorRamp
Color
Alpha
RGB
Linear
1
Pos
0.886
Fac
terracotta_height.png
Color
Alpha
terracotta_height...
Linear
Flat
Repeat
Single Image
Color Space
Non-Color
Vector
RGB to BW
Val
Color
terracotta_normal.png
Color
Alpha
terracotta_normal...
Linear
Flat
Repeat
Single Image
Color Space
Non-Color
Vector
Normal Map
Normal
Tangent Space
UVMap
Strength
8.200
Color
Bump
Normal
Invert
Strength
0.600
Distance
4.400
Height
Normal
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
Subsurface Colo
Metallic
0.000
Specular
0.500
Specular Tint
0.000
Roughness
Anisotropic
0.000
Anisotropic Rotation
0.000
Sheen
0.000
Sheen Tint
0.500
Clearcoat
0.000
Clearcoat Roughness
0.030
IOR
1.450
Transmission
0.000
Transmission Roughness
0.000
Emission
Emission Strength
1.000
Alpha
1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement

Figure 17.29

Node Wrangler

A quick way to set up a BBR Node Arrangement is to use Blender's **Node Wrangler.** This is an **Add-on** included with the program which you activate in the **Preferences Editor** (Reference Chapter 2 -2.17).

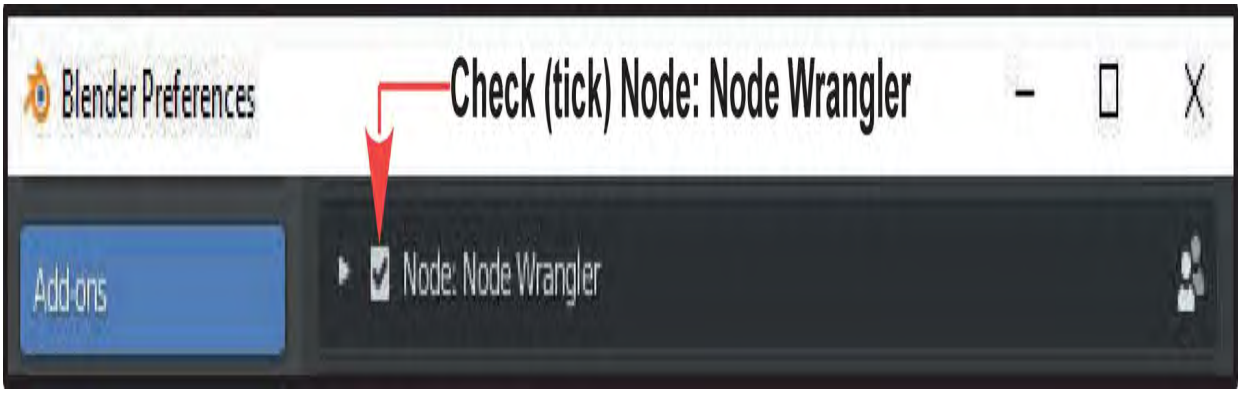


Figure 17.30

Have the PBR Texture Images saved on your computer.

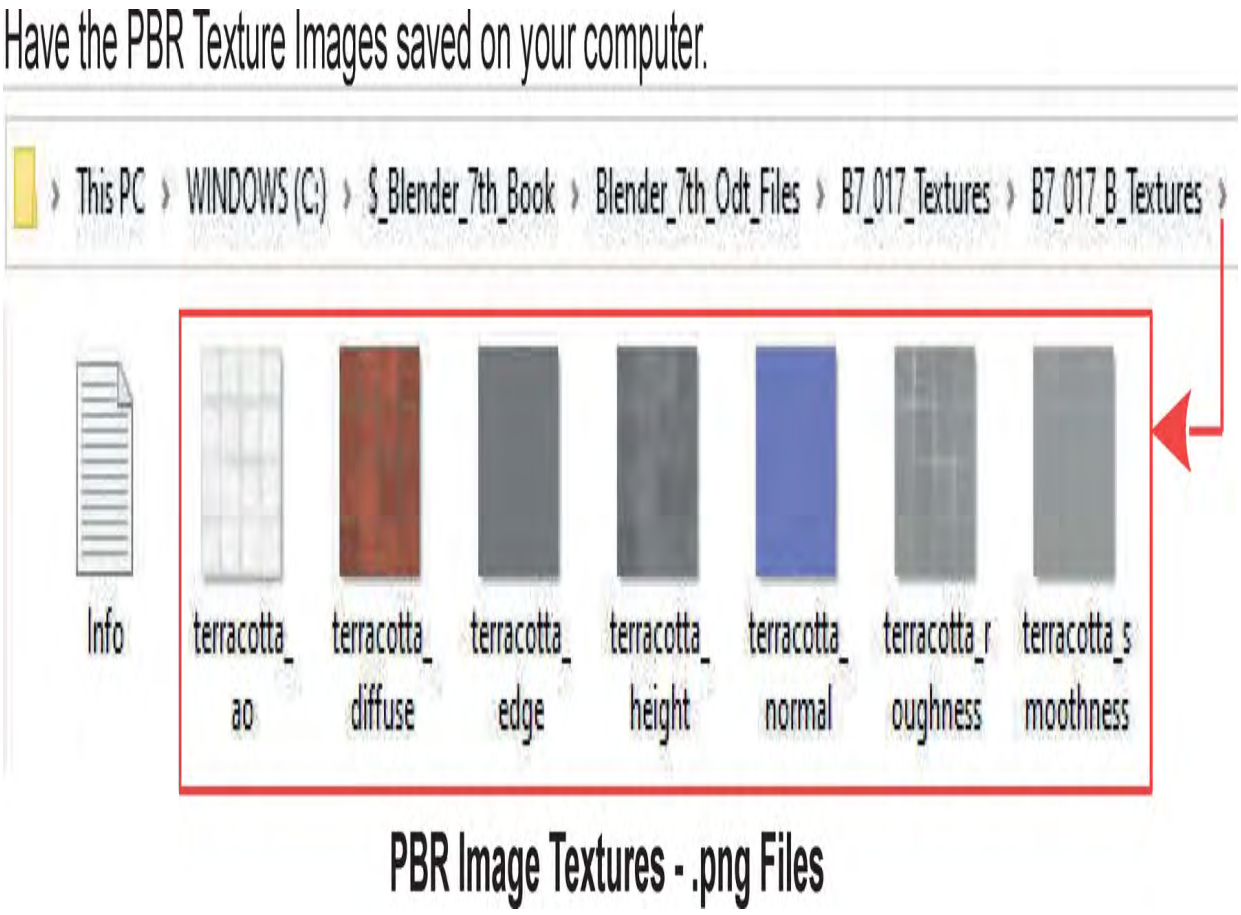


Figure 17.31

In the 3D Viewport Editor have the Object to which the Texture is to be applied selected. Remember; the Object must have a Material applied (default gray is fine).

In the Shader Editor you will see the Principled BSDF Node connected to the Material Output Node which produces the default gray Material on the selected Object.

In the Shader Editor, mouse over on the Principled BSDF Node then press **Ctrl + Shift + T Key.** The Blender File View displays. Navigate to the Folder containing the PBR Texture Images (.png).

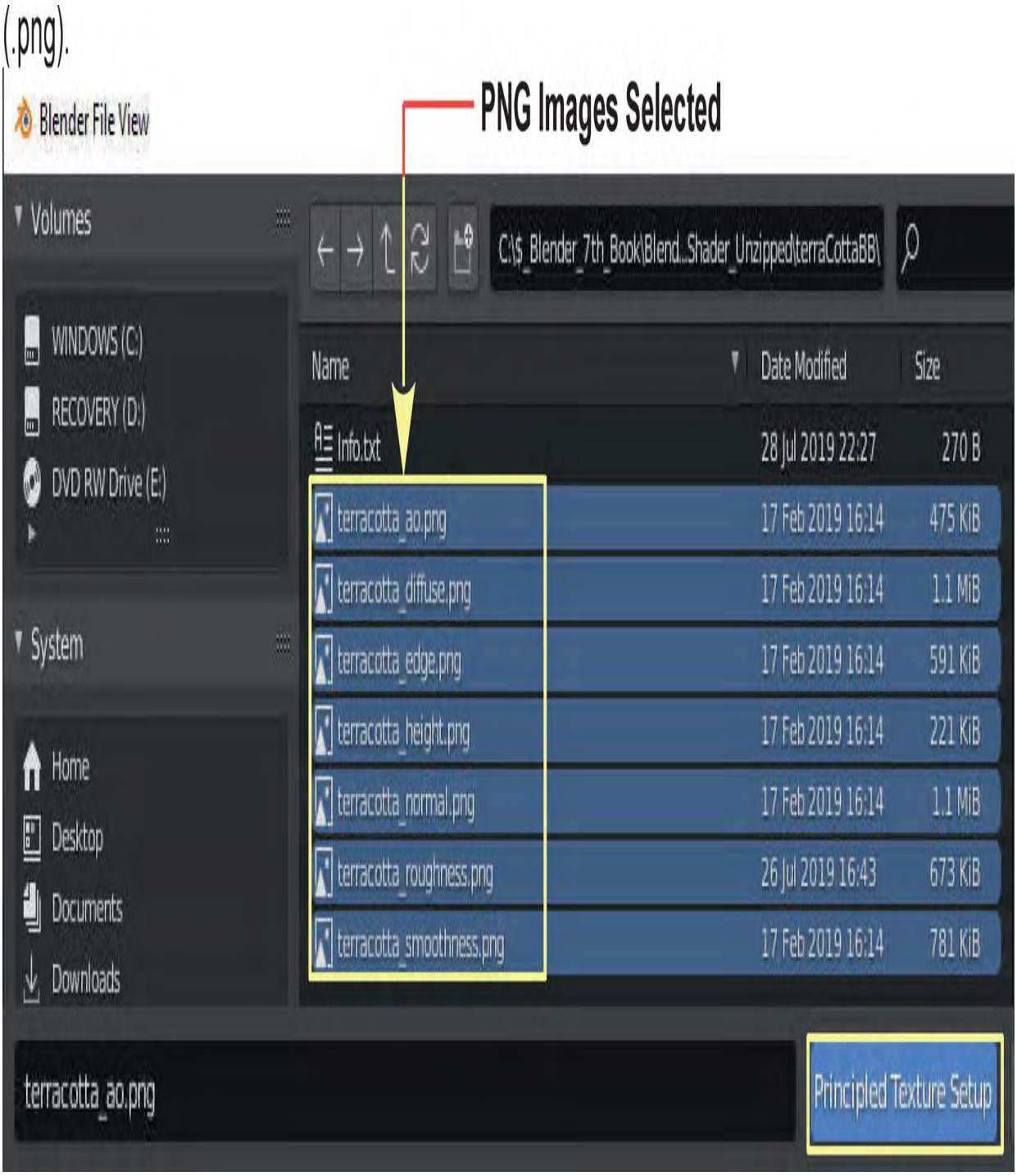


Figure 17.32

Shift select the Texture Files you wish to use then click **Principled Texture Setup** at the bottom of the panel (Figure 17.32).

A **PBR Texture Node Arrangement** is automatically generated in the **Shader Editor** with the result exported to the Object in the 3D Viewport Editor (Figure 17.33)

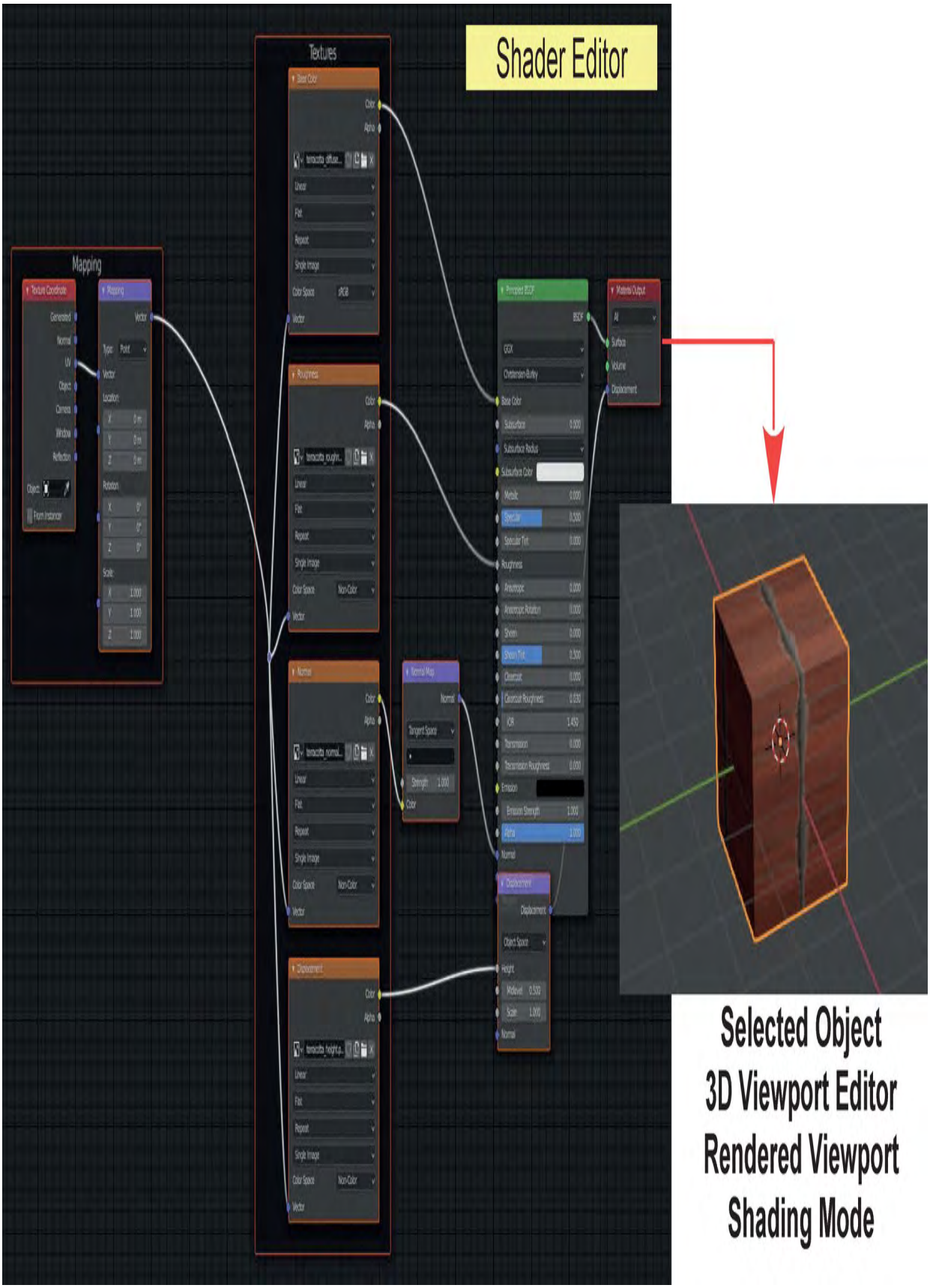

Shader Editor
Textures
Mapping
Selected Object
3D Viewport Editor
Rendered Viewport
Shading Mode

Figure 17.33

17.6 Texture Properties:

Textures are applied to an Object's surface in the **Material Properties** and configured in the **Shader Editor.** The **Texture Properties** panel in the **Properties Editor** is used as **Texture Repository** allowing quick access to Image Textures for assignment to Objects in the 3D Viewport Editor.

To demonstrate, have several Images save on your computer to be used as Image Textures.

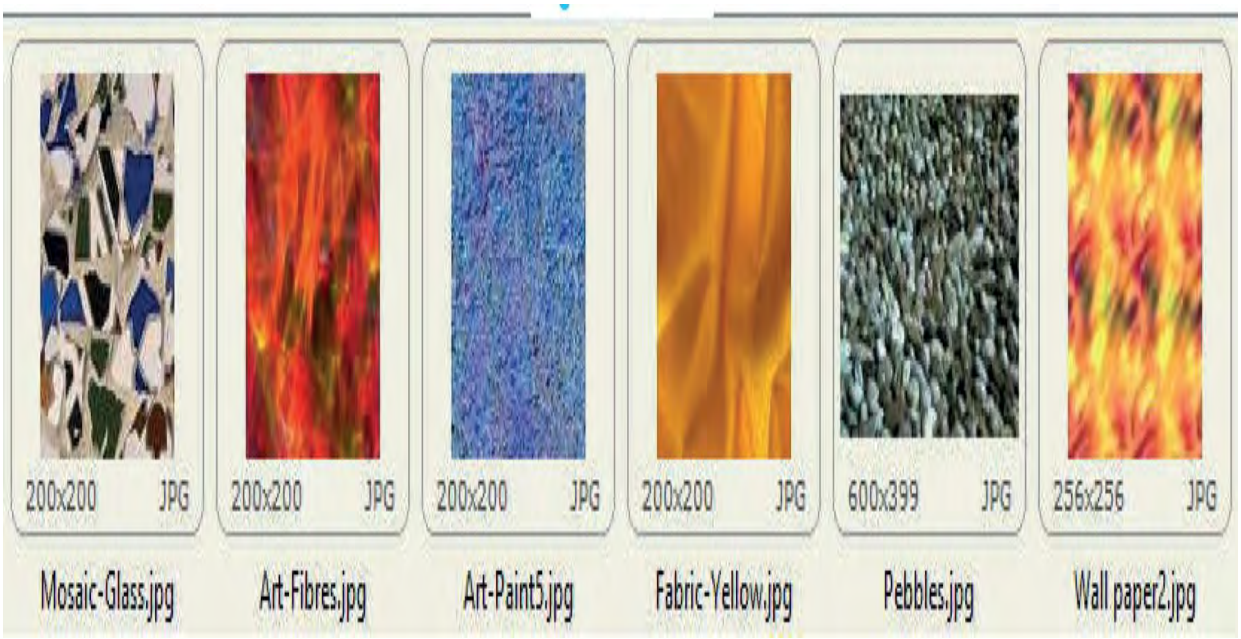
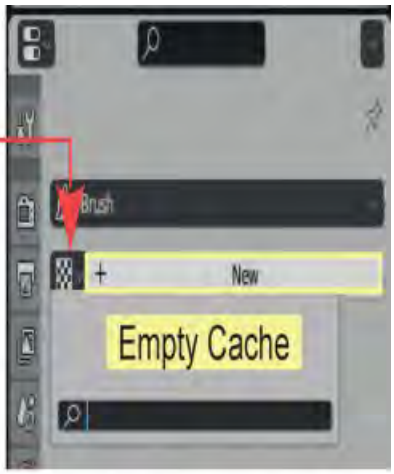


Figure 17.34

In the **Properties Editor, Texture Properties** click the **New Button.**

Note: Clicking the **Browse Texture to be Linked button** reveals that there are no Textures for selection (Empty Cache).



Clicking the **New button** displays the contents of the Texture Properties panel and you see that a **Texture Channel** has been entered which by clicking the Browse Texture to be Linked button shows a Texture named **Texture** in the Cache.

In the **Settings Tab** click the **Open button** and navigate to the first of your images to be used as Textures. Click **Open Image**. The Image displays in the Preview panel and the Cache will show that the Image has been assigned to the **Texture Channel.**

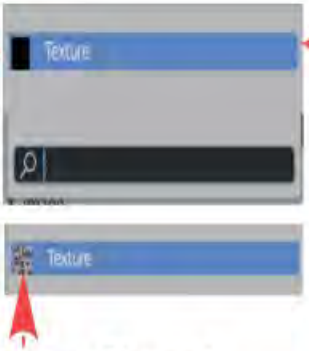


Figure 17.35

Note: There does not have to be an Object in the 3D Viewport Editor at this point. As proof of this delete the default Cube.

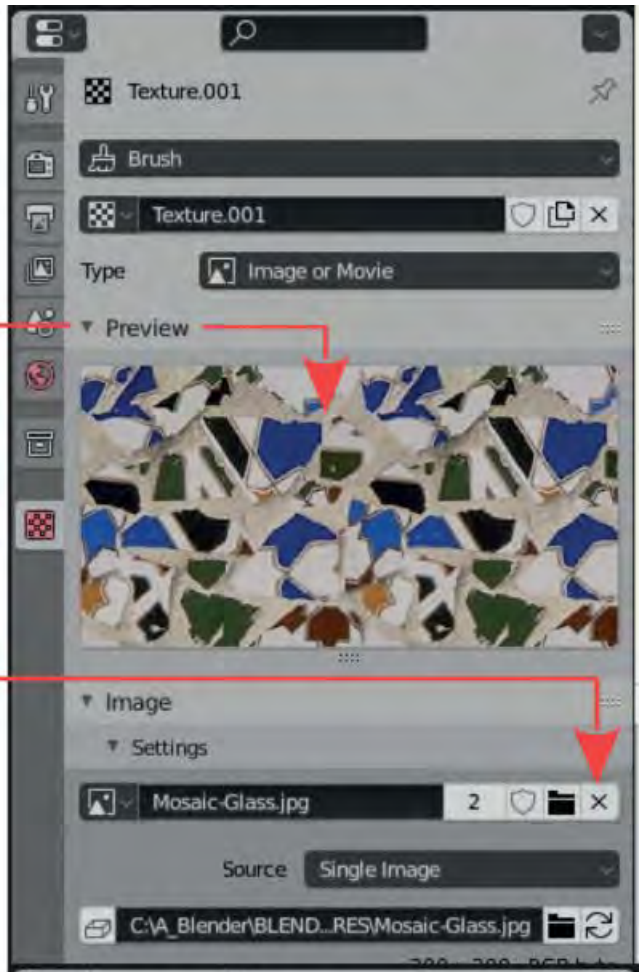
Note: The thumbnail in the Texture Channel may take a second to load in the Preview.

Click the **Add New Texture button** a second time creating a new texture Channel named **Texture.001.** At this point the channels named Texture and Texture.001 both have the same Image Data assigned as shown in the Preview Panel, therefore, with **Texture.001** selected click the **Unlink data block button** in the **Image - Settings.**

With the data unlinked click the **Open button** in the **Image — Settings,** navigate and select your second Texture Image.

Preview panel showing Image Data assigned to Texture and Texture.001

Unlink data block button



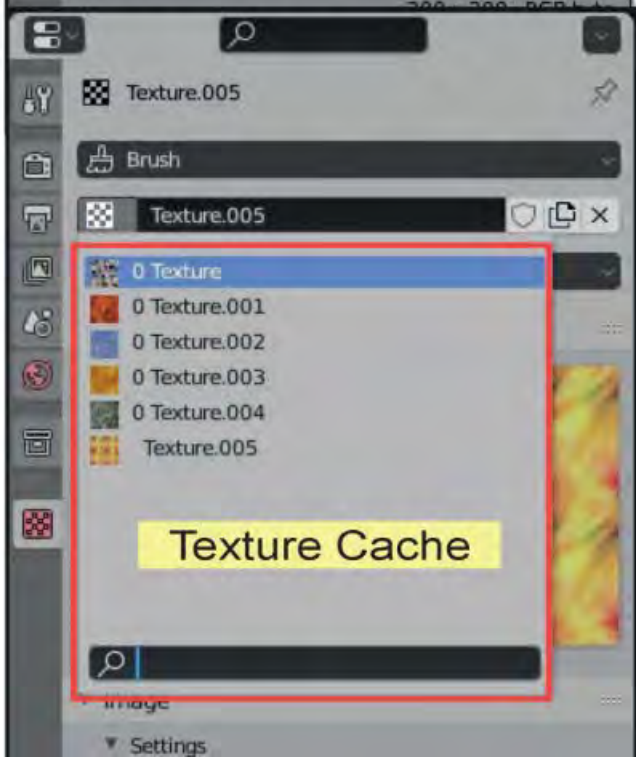


Figure 17.37

You now have Texture with Image 1 assigned and Texture.001 with Image 2 assigned.

Repeat the procedure for the remaining Image textures.

The **Texture Cache** will display with textures which you can assign to any Object which is selected in the 3D Viewport Editor (Figure 17.38 - Lower).

To apply a Texture to an Object, select the Object in the 3D Viewport Editor and **apply a Material.** In the **Properties Editor, Material Properties** click the **Base Color button,** select **Image Texture** in the menu that displays.

In the **Shader Editor** you have an Image Texture Node connected to the Base Color input socket of the Principled BSDF Node. In the **Image Texture Node,** click the **Browse Image to be Linked** button and select one of the Textures from the Cache.

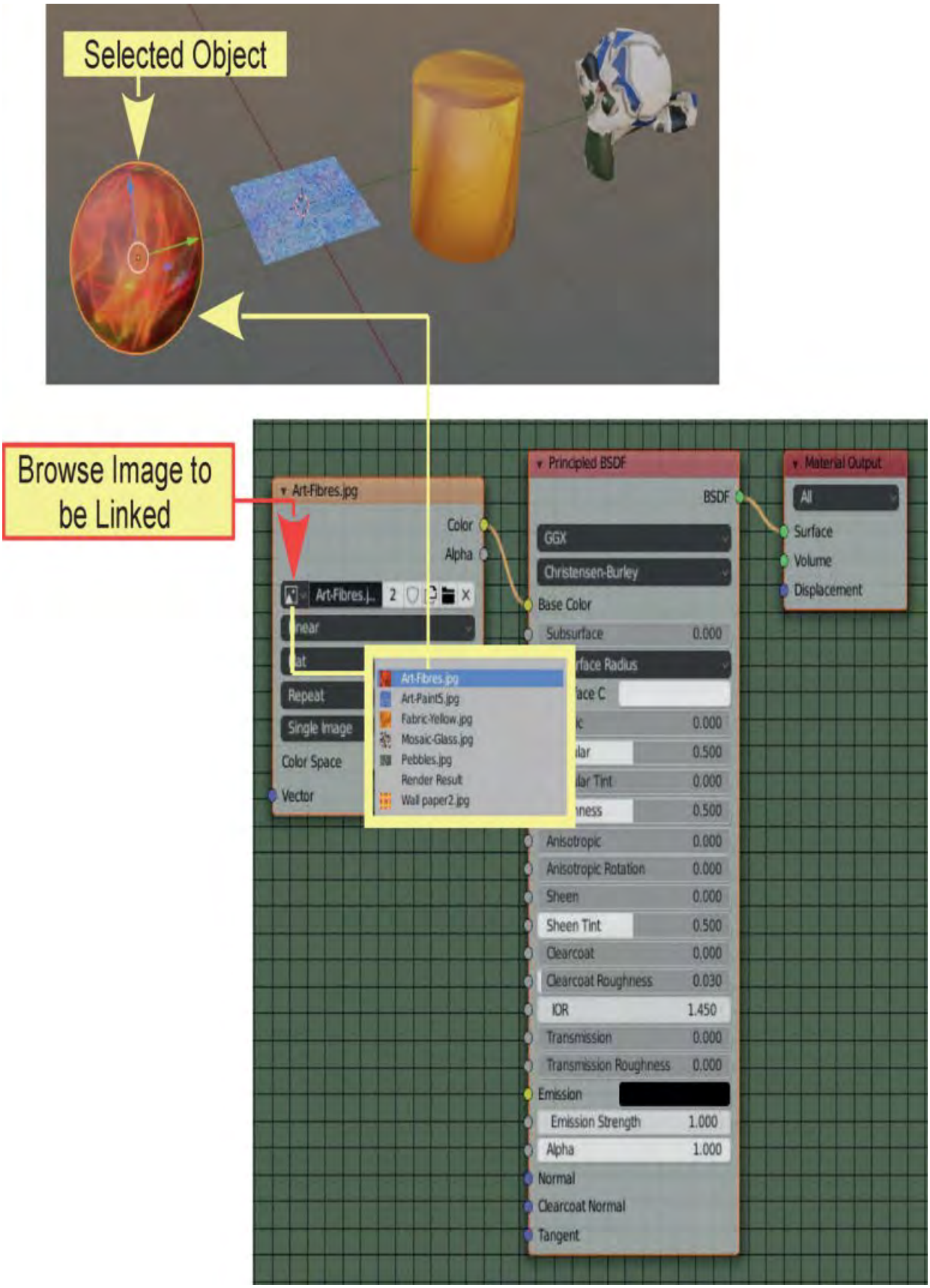

Selected Object
Browse Image to be Linked
Art-Fibres.jpg
Color
Alpha
Art-Fibres.j...
2
Repeat
Single Image
Color Space
Vector
Art-Fibres.jpg
Art-Paint5.jpg
Fabric-Yellow.jpg
Mosaic-Glass.jpg
Pebbles.jpg
Render Result
Wall paper2.jpg
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement

Figure 17.38

17.7 Texture Painting

Texture Painting is the process of modifying a Texture that has been Mapped to the surface of an Object by Painting over the Texture with a Material Color or a different Texture saved in a Blender Texture File Cache.

To demonstrate the process a Texture will be applied to Suzanne (Monkey Object) and painted over to make her beautiful ("beauty is in the eye of the beholder").

In a new Blender file delete the default Cube Object and add a Monkey Object. Rotate Suzanne in the 3D Viewport Editor to display her face. By default Susanne displays with her surface features made up from a series of flat planes which you see when she is placed in Edit Mode. This display will suit the purpose for the Painting demonstration.

Have Susanne in Object Mode in the 3D Viewport Editor. The process being demonstrated is painting over a texture (Texture Painting), therefore, the first step is to apply a Texture to Susanne.

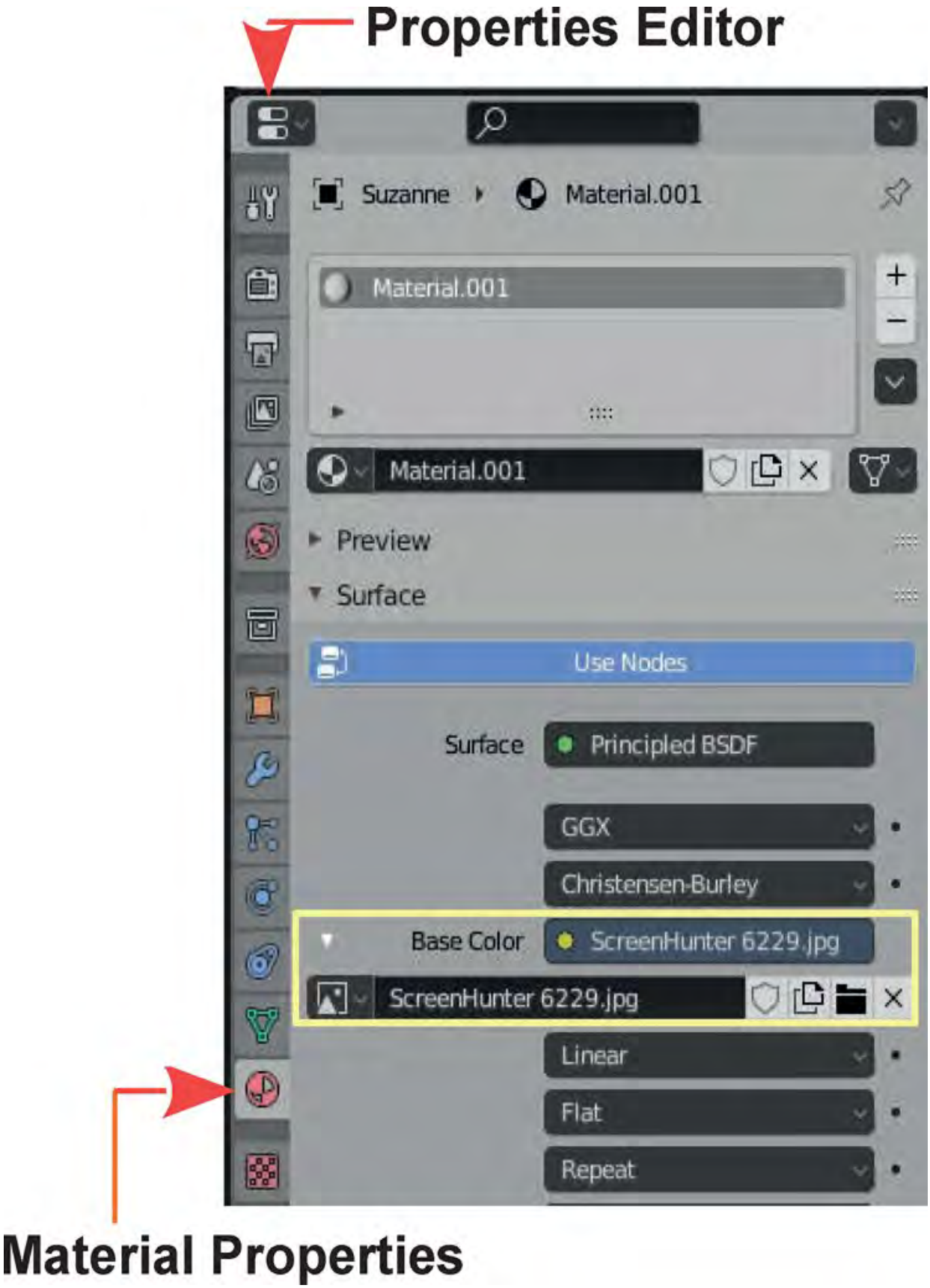

Properties Editor
Suzanne
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
ScreenHunter 6229.jpg
ScreenHunter 6229.jpg
Linear
Flat
Repeat
Material Properties

Figure 17.39

A Texture is applied in the Properties Editor, Material Properties which enforces the golden rule that a Material must be added before a Texture. Remember; the default 3D Viewport displays in Solid Viewport Shading Mode and Textures will only display in Material Preview or Rendered Viewport Shading Modes.

Since the Texture is to be painted over it is best to use a simple plain bland Image Texture (reference 17.3. For this demonstration the JPEG Image shown in Figure 17.40 will be used which has been made by adding a bland Material (color) to a Plane Object and taking a Screen Capture.

Figure 17.40

Suzanne in the 3D Viewport Editor

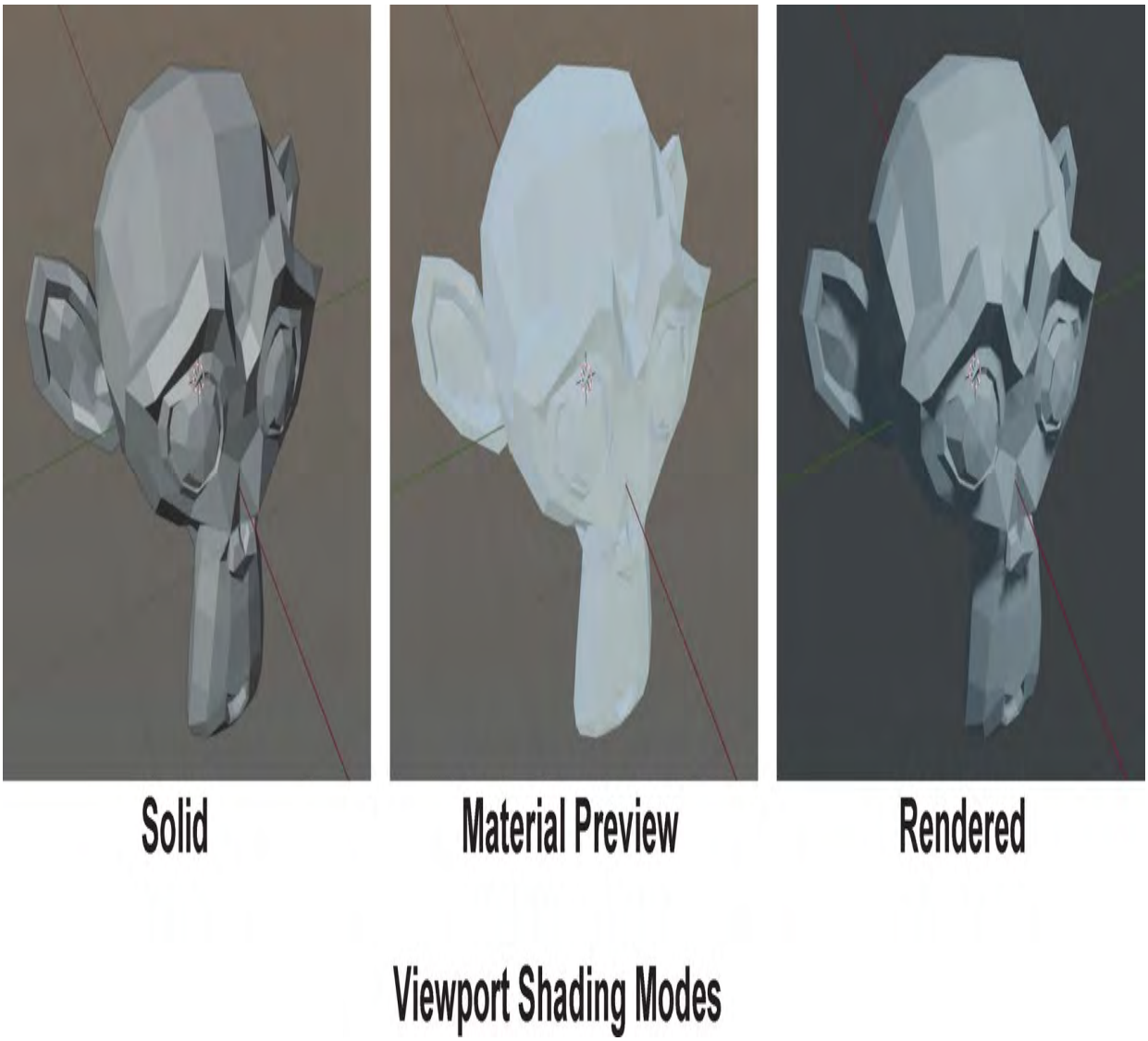


Figure 17.41

Change the default **Layout Workspace** to the **Texture Paint Workspace** and make sure the **3D Viewport Editor** is in **Texture Paint Mode**.

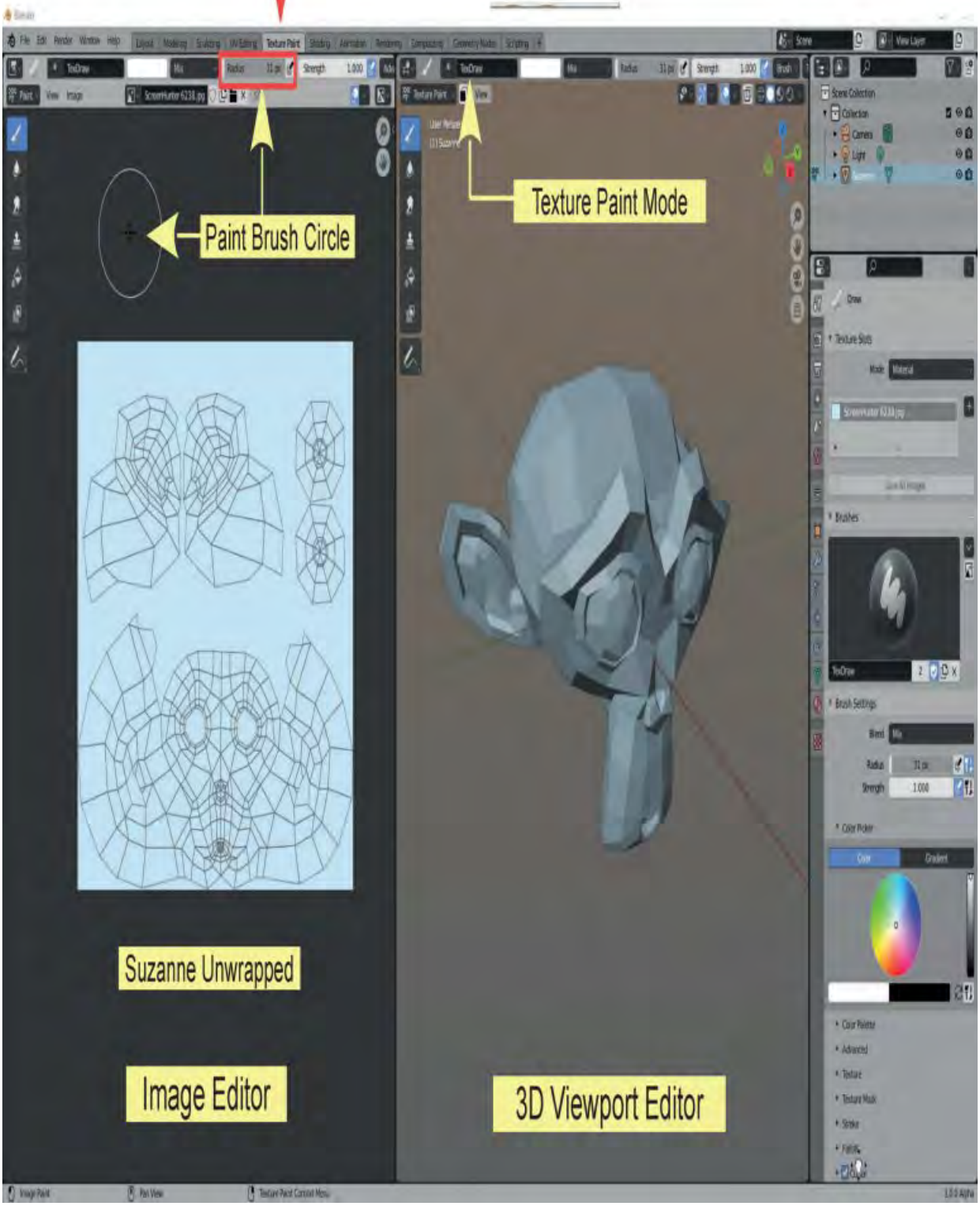


Figure 17.42

The **Texture Paint Workspace** displays with the **Image Editor** in **Paint Mode** and the **3D Viewport Editor** in **Texture Paint Mode.** In the Image Editor Suzanne has been automatically unwrapped laying out the surfaces on a flat plane.

Note: The 3D Viewport Editor is in Solid Viewport Shading Mode. To see Textures change this display to Material Preview or Rendered Viewport Shading.

The flat plane in the Image Editor and the surface of Suzanne in the 3D Viewport Editor display with the Texture (Screen Hunter 6229.jpg) that has been applied in the first step of the process.

At this point you can Paint, in the Image Editor over the Texture or in the 3D Viewport Editor. You Paint by clicking, holding and dragging the Brush Circle. The intention is; to Paint in specific Faces in the Image Editor which will Paint on to the corresponding Face on the surface of Suzanne, therefore, reduce the Brush Circle diameter to 5 px or 6 px in the Image Editor Header.

Figure 17.43

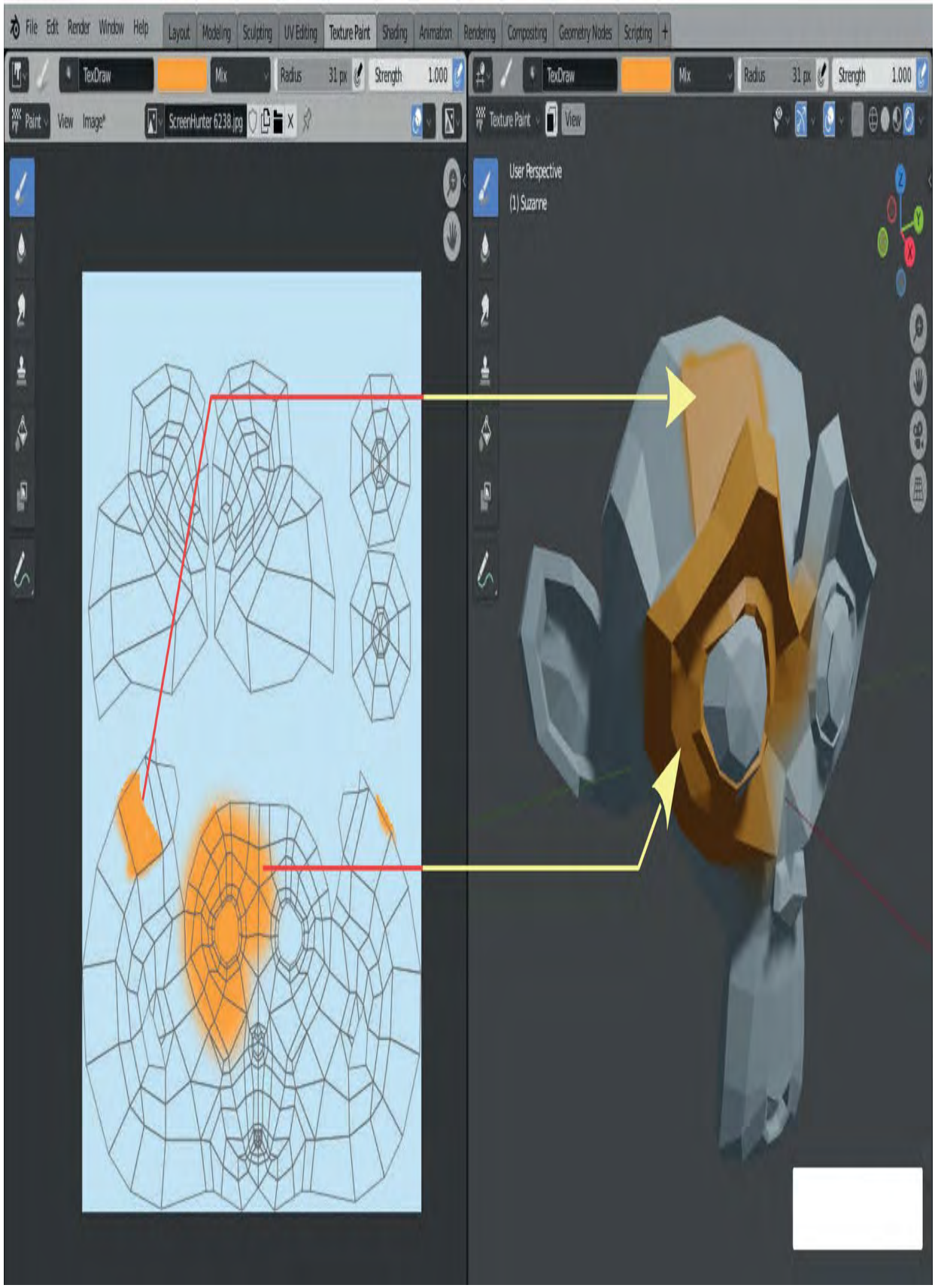

File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Geometry Nodes Scripting
TexDraw
Mix
Radius 31 px
Strength 1.000
Paint View Image*
ScreenHunter 6238.jpg
Texture Paint View
User Perspective
(1) Suzanne

Figure 17.44

Tip: To adjust the Brush Circle diameter hold the F Key and drag the Mouse.

Painting with a Texture

Painting a Material color is painting the color over the Texture that has been applied to Susanne (the Monkey Object). Susanne is also the selected Object in the 3D Viewport Editor. With multiple Objects in a Scene the Texture Paint Mode is applicable to the Selected Object.

Besides painting a Material Color you may also paint with a Texture (Paint a Texture over the applied Texture).

The Texture is placed in the **Texture Cache**.

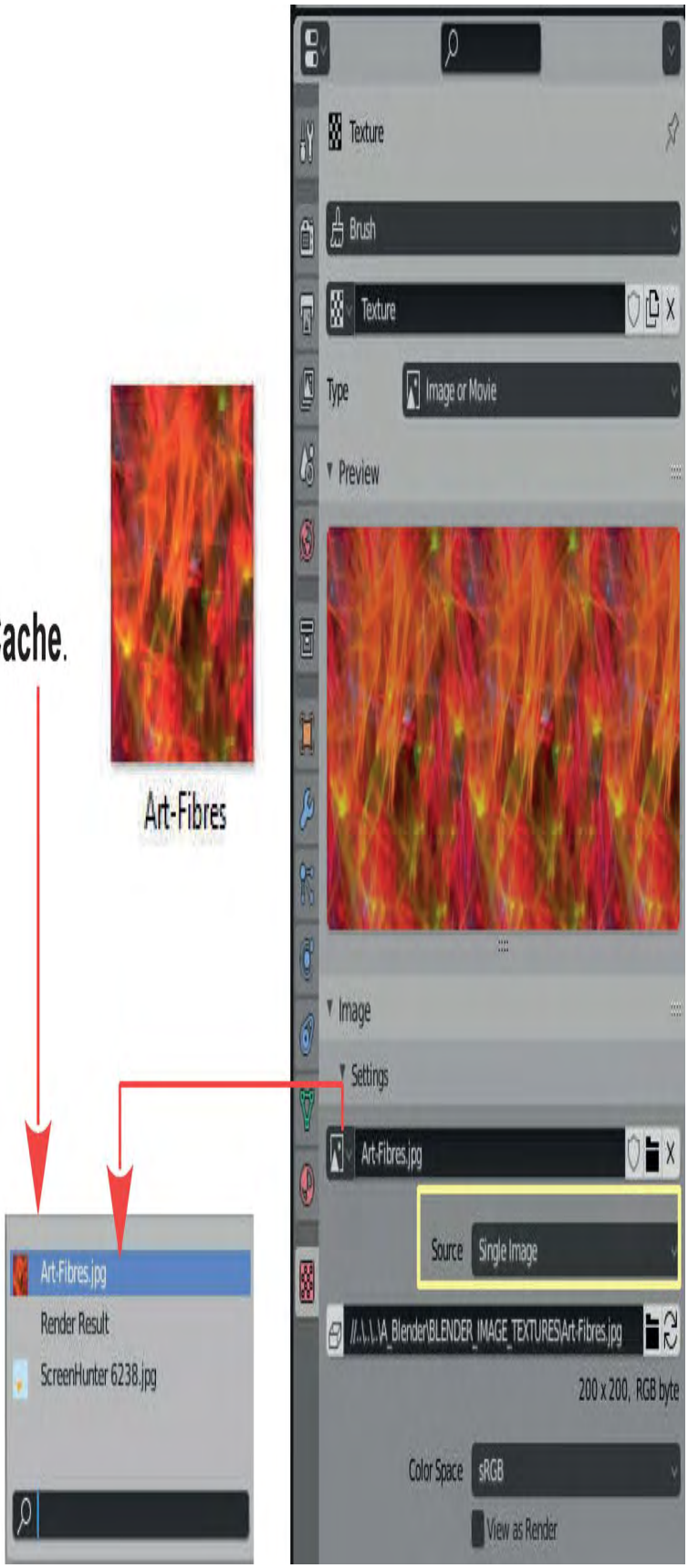


Figure 17.45

Create a Texture in the Properties Editor, Texture Properties (reference Section 17.7).

In the example a Texture has been generated using a JPEG Image named Art-Fibres.

Note: Source: **Single Image.**

With the Texture created and placed in the Cache you can paint the Texture in the Image Editor or in the 3D Viewport Editor.

Blending Material Color with Textur

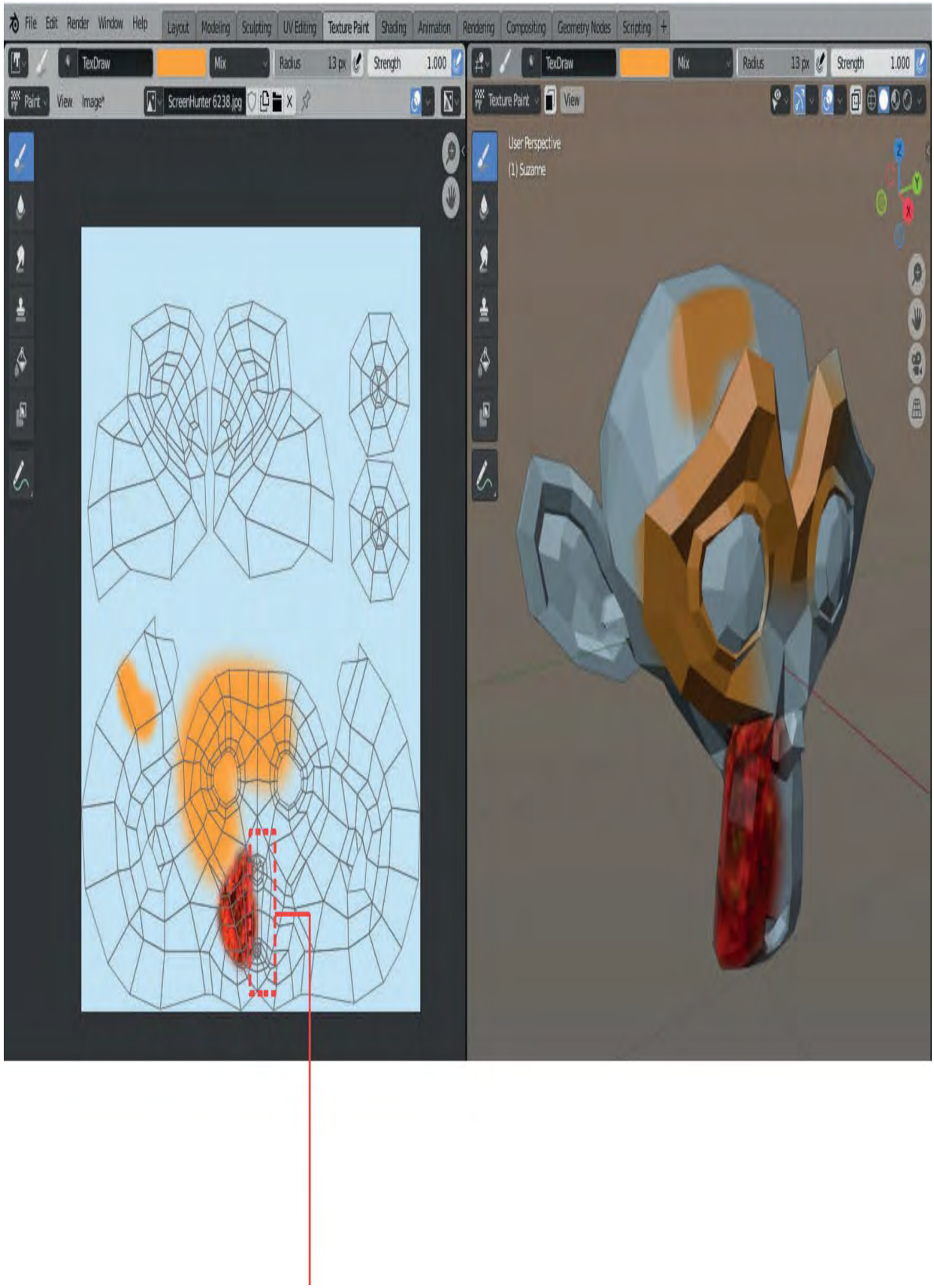

File Edit Render Window Help
Layout Modeling Sculpting UV Editing Texture Paint Shading Animation Rendering Compositing Geometry Nodes Scripting
TexDraw
Mix
Radius 13 px
Strength 1.000
Paint View Image*
ScreenHunter 6238.jpg
Texture Paint View
User Perspective
(1) Suzanne

Figure 17.46

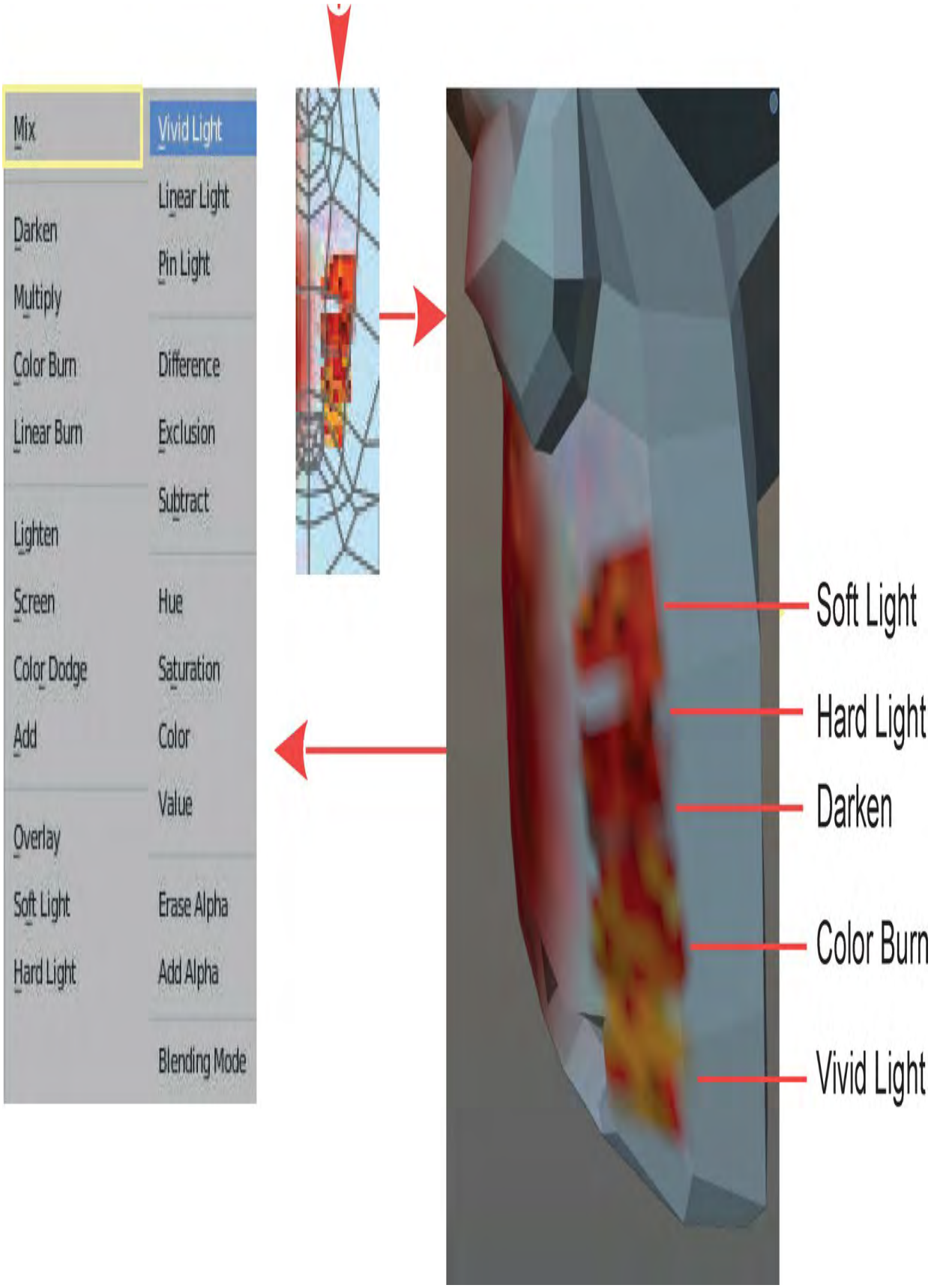

Mix
Darken
Multiply
Color Burn
Linear Burn
Lighten
Screen
Color Dodge
Add
Overlay
Soft Light
Hard Light
Vivid Light
Linear Light
Pin Light
Difference
Exclusion
Subtract
Hue
Saturation
Color
Value
Erase Alpha
Add Alpha
Blending Mode
Soft Light
Hard Light
Darken
Color Burn
Vivid Light

Figure 17.47

17.8 Texture Displacement

In **Chapter 9 — 9.6 Displace Modifier,** a simple demonstration showed how to displace a Vertex Group, using the Displace Modifier with one of Blenders **Procedural Textures** (In Built Texture). Vertices were displaced according to the light and dark components of the Texture.

Image Textures may also be used, in which case, displacement is determined by the color shades in the Image. The Vertex displacement may be applied to the entire Mesh surface not only a Vertex Group.

Prerequisites:	**1.** You must have an Image saved on your PC to be used as a Texture. To demonstrate Texture Displacement the image shown in Figure 17.48 will be used. **2.** Have the surface Mesh (Plane Object) subdivided creating plenty of Vertices.

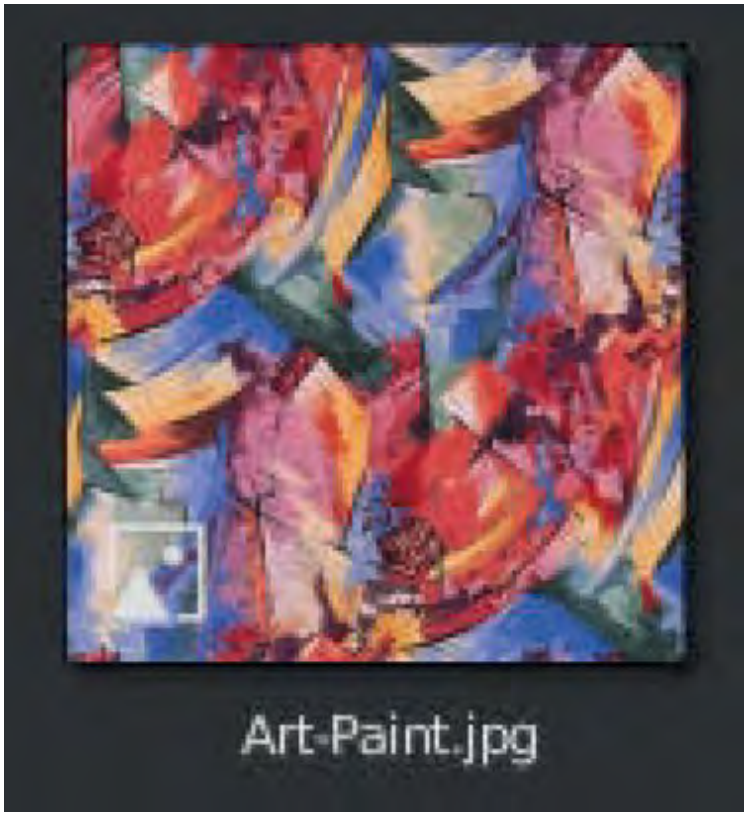


Figure 17.48

Note: Images may be used as a Texture to simply color a surface, in which case, the image is displayed on the surface of the mesh (Ref. Section 17.3).

Remember: To see color, be in **Material Preview** or **Rendered** Viewport shading Mode.

Coloring using Texture.

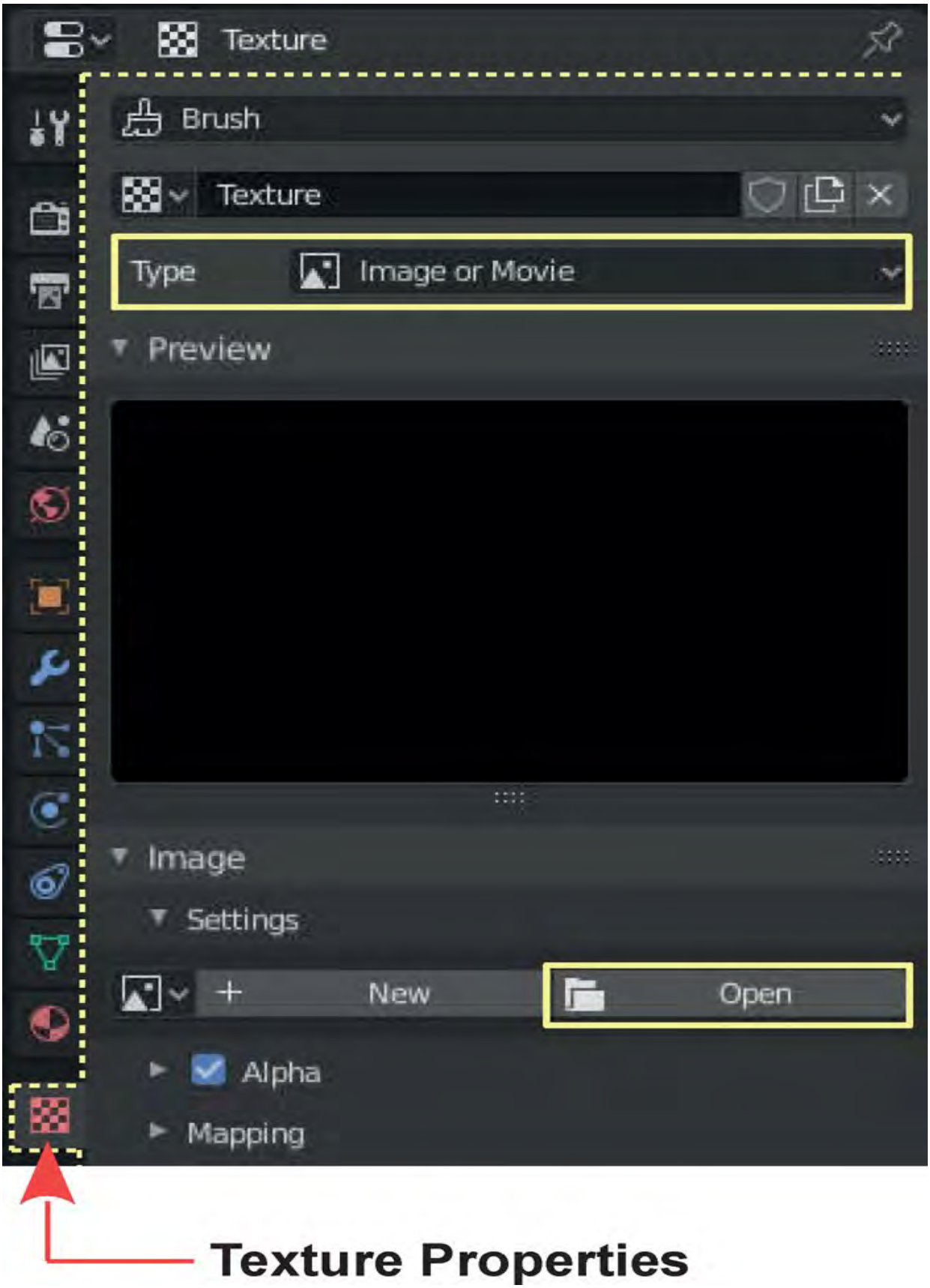

Texture
Brush
Texture
Type
Image or Movie
Preview
Image
Settings
New
Open
Alpha
Mapping
Texture Properties

Figure 17.49

With the Object in the 3D Viewport Editor selected, **Open** the **Material Properties** in the Properties Editor with **Use Nodes** active. Click on the button at the end of the **Base Color** bar and select **Image Texture** from the menu. The Base Color will show **Image Texture** in what was the color bar. Click on **Open** and navigate in the File Browser to find your image file. Select a file and click **Open Image.** In Material Preview or Rendered Viewport Shading Mode the Texture Image is mapped to the surface of the Object.

Displacement using Texture

The displacement is performed by entering a Texture in the **Displace Modifier** (Ref: Chapter 9 — 9.6). To enter a Texture in the Modifier the Texture must first be entered in the **Texture Cache.**

In the **Properties Editor, Texture Properties** click **New.** The buttons display with **Type Image or Movie** (Figure 17.49). In the Image Tab, Settings, click **Open** (the File Browser opens). Navigate to your Texture Image and open it. This enters the Texture in the Texture Cache (Ref. 17.7 Texture Properties).

Note: To use a texture for surface displacement it is not necessary to have it entered as a Material. The surface deformation will still occur in accordance with the color shading of the image but the colors do not show on the deformed surface. If you wish to display the colors enter the Texture Image in the Material Properties as previously described.

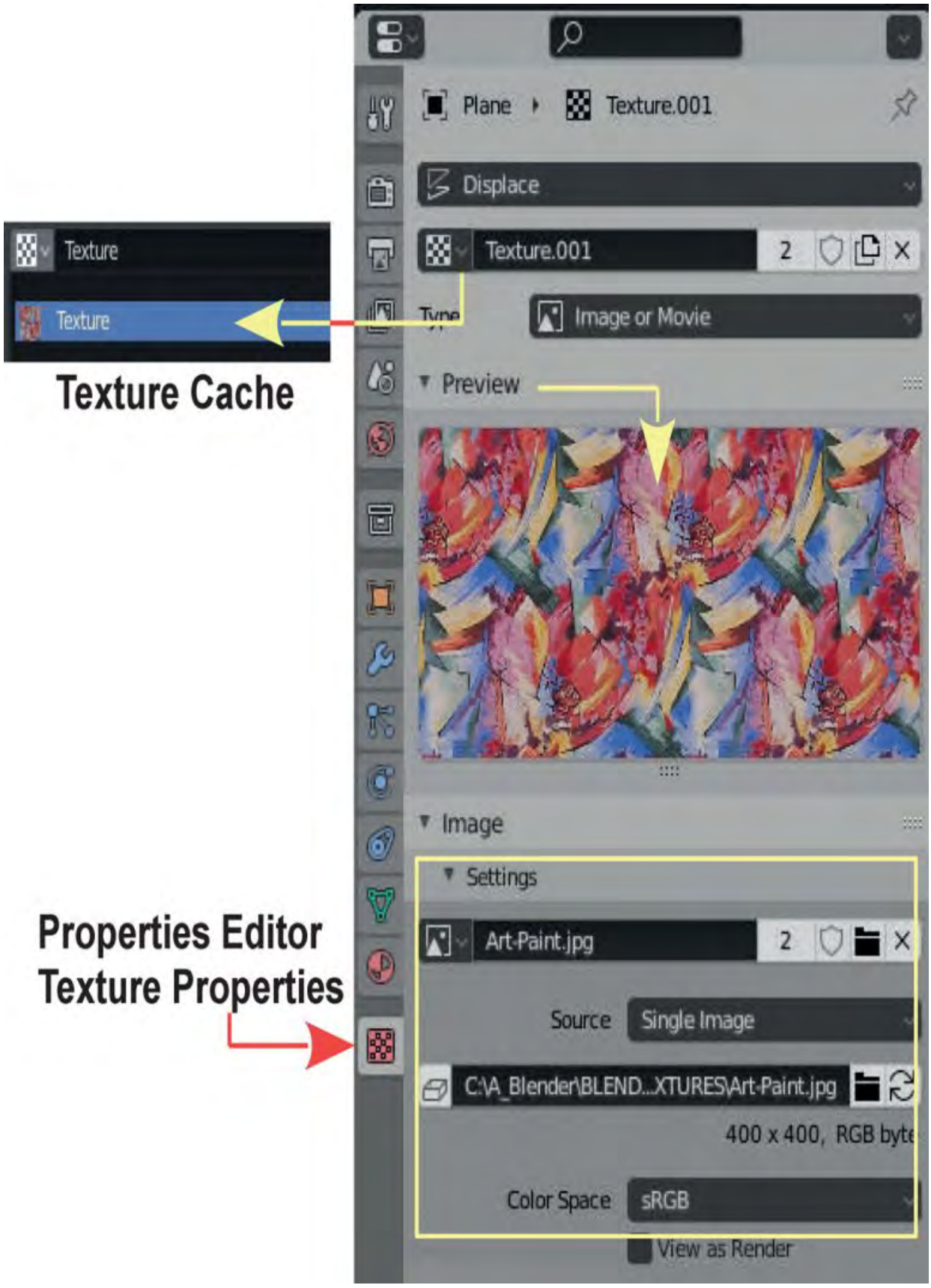

Texture
Texture
Texture Cache
Plane
Texture.001
Displace
Texture.001
2
Type
Image or Movie
Preview
Image
Settings
Art-Paint.jpg
2
Source
Single Image
C:\A_Blender\BLEND...XTURES\Art-Paint.jpg
400 x 400, RGB byte
Color Space
sRGB
View as Render
Properties Editor
Texture Properties

Figure 17.50

With the Texture Image entered in the texture Cache have the Object surface selected in the 3D Viewport Editor and add a **Displace Modifier** (Figure 17.51). In the Displace Modifier click on the **Browse Texture to be Linked button** to display the Texture Cache and select the Texture you previously entered (Figure 17.50).

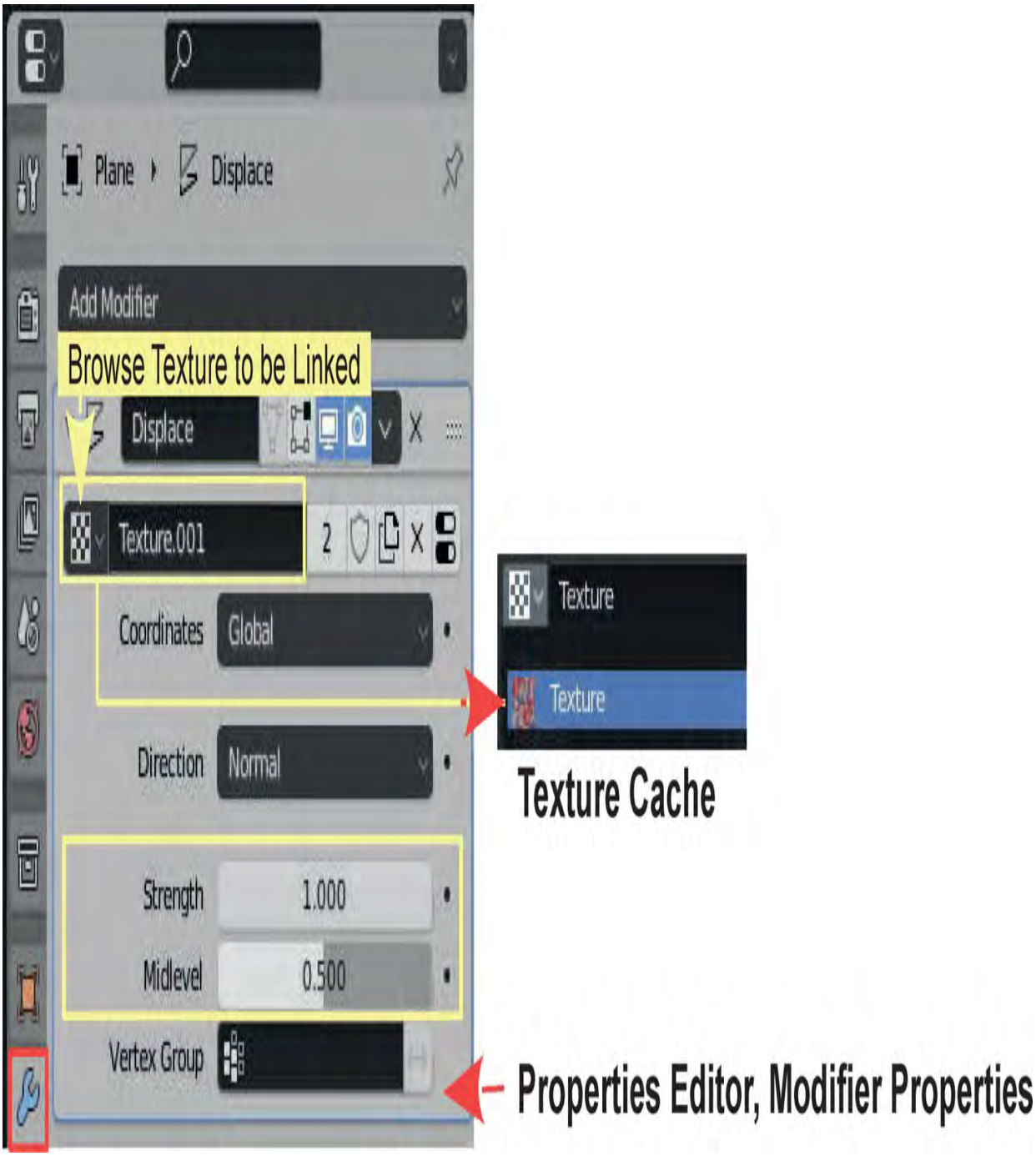


Figure 17.51

The Objects Mesh surface is deformed (Figure 17.52). The amount of deformation is controlled using the **Midlevel** and **Strength** sliders in the Modifier panel.

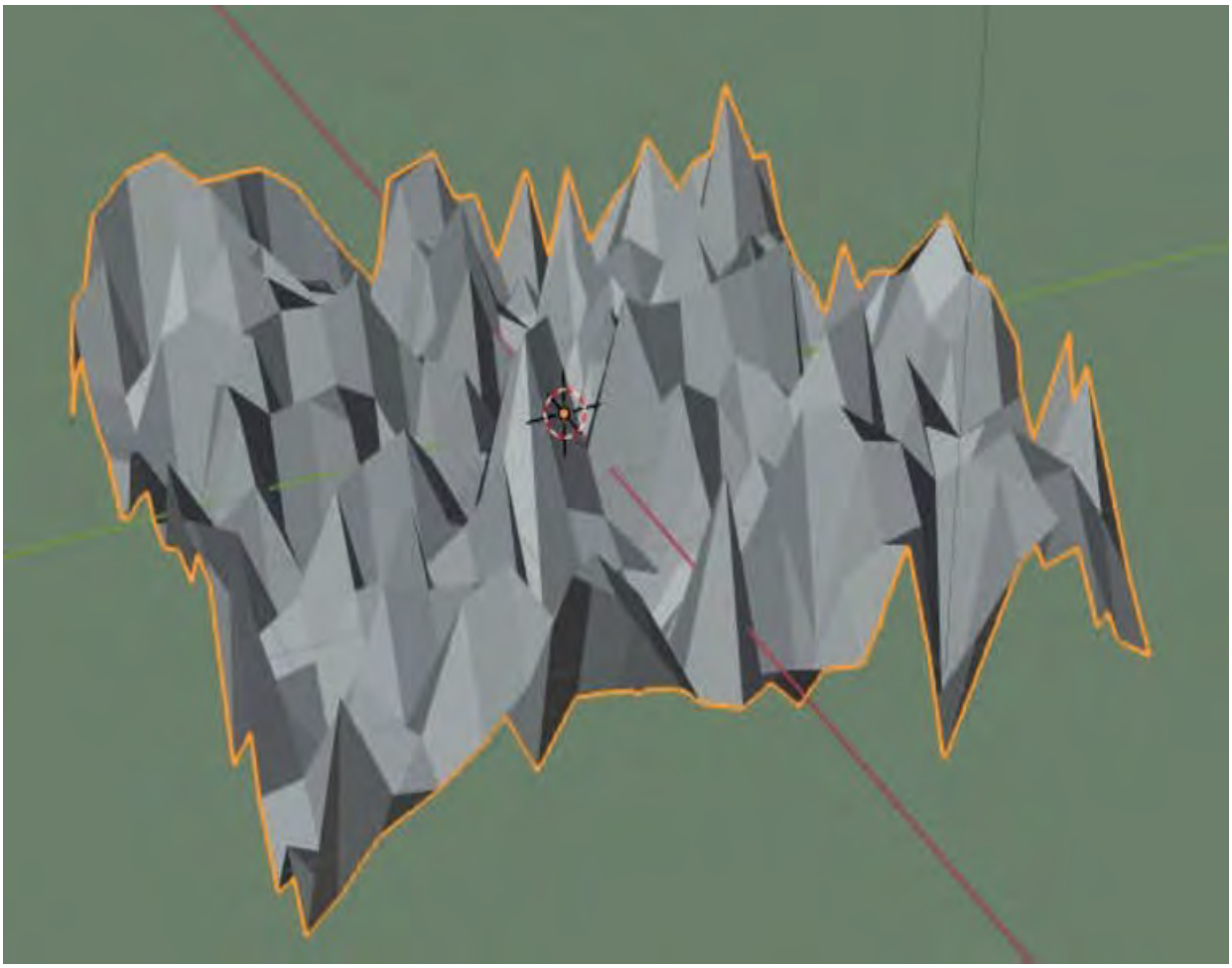


Figure 17.52

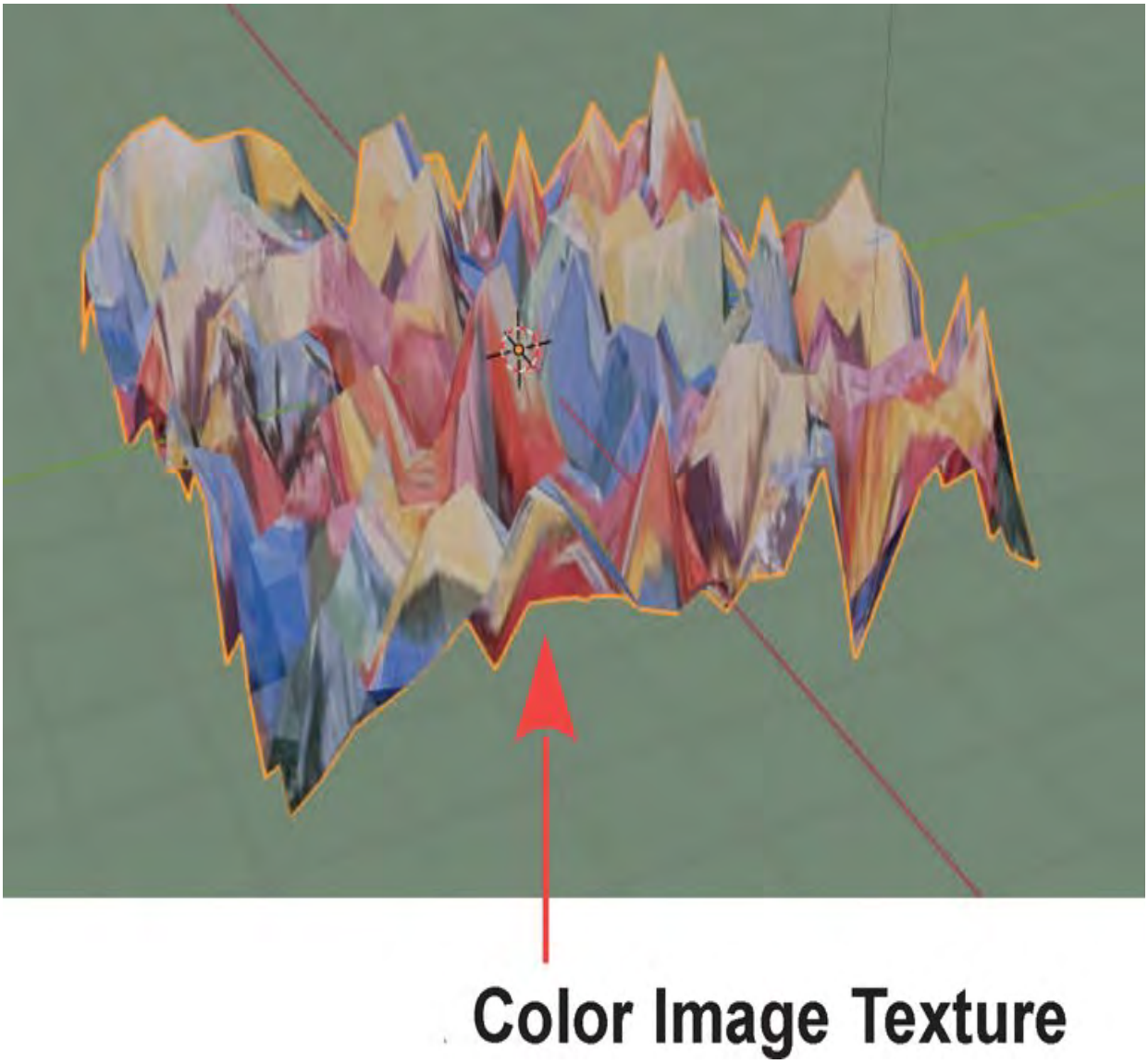


Figure 17.53

18 Node Systems and Usage

18.1 The Texture Node Editor
18.2 The Geometry Node Editor
18.3 Transparency
18.4 Volumetric Lighting
18.5 Compositing Nodes

Since Nodes are integral when applying Materials and Textures they have been introduced in the context of those subjects. There are, however, many uses for Node Systems. The following examples will show a small sample with the **Eevee Render System** engaged. The Cycles Render System also uses Nodes although the Node selection differs.

Nodes are a graphical representation of code processing blocks which may be arranged and connected to produce a variety of effects. The Blender interface may be configured to display four (4) different **Node Systems** as seen in the **Viewport Editor Type**, selection menu.

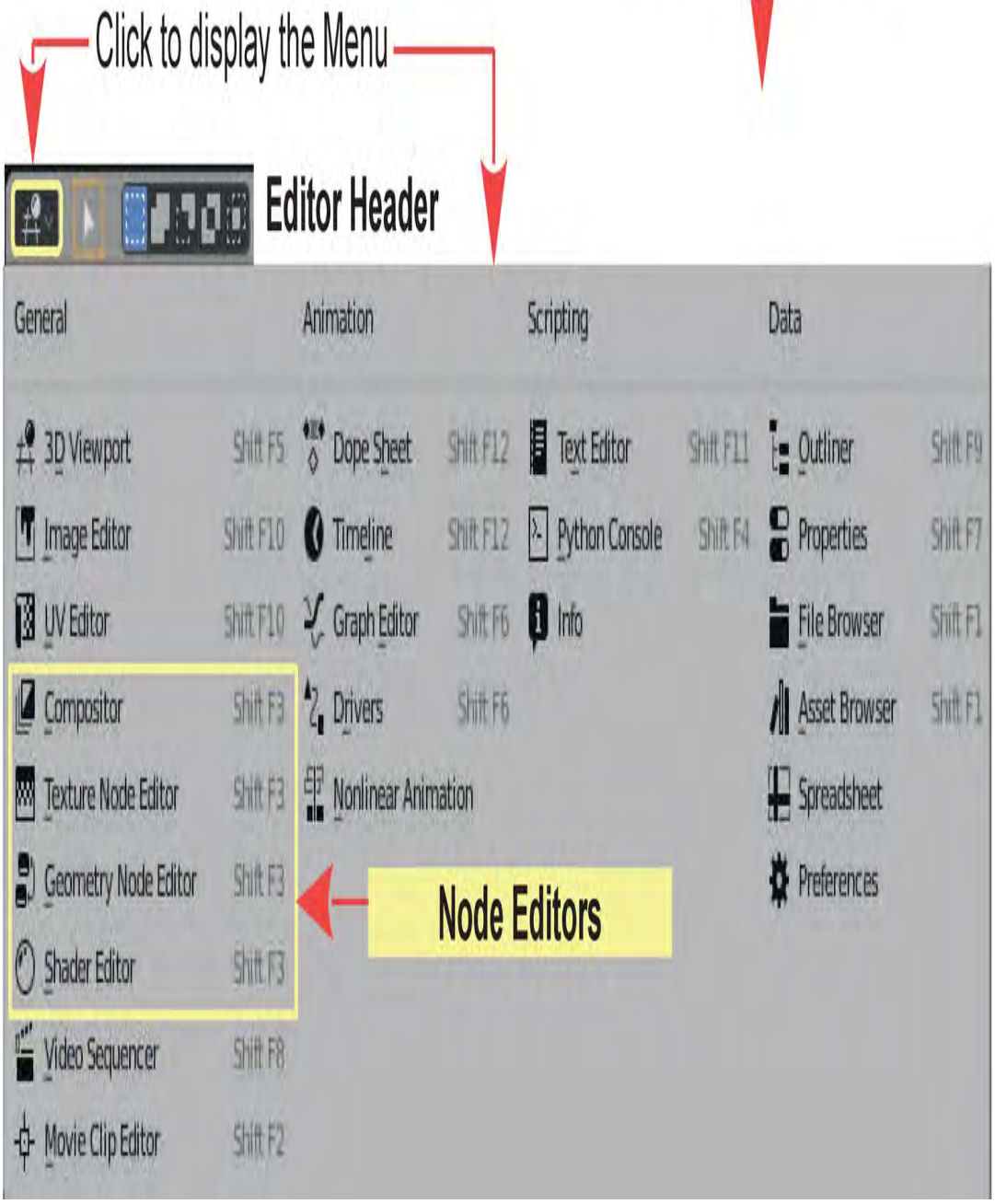


Figure 18.1

Up to this point Nodes have been discussed only in the **Shader Editor** when considering the application of Materials and Textures. The Textures were described as Material Textures since they were applied in the **Properties Editor, Material Properties** using the **Shader Editor.**

18.1 The Texture Node Editor

The **Texture Node Editor** is accessed from the **Editor Type Selection Menu** (Figure 18.1). To introduce the Texture Node Editor the **Texture Paint** operation in the 3D Viewport Editor will be demonstrated.

In the default Screen arrangement, drag the upper edge of the Timeline Editor up and change the Timeline Editor to the **Texture Node Editor.** Put the Editor in **Brush Mode** (Figure 18.2).

Have the 3D Viewport Editor in **Texture Paint Mode.**

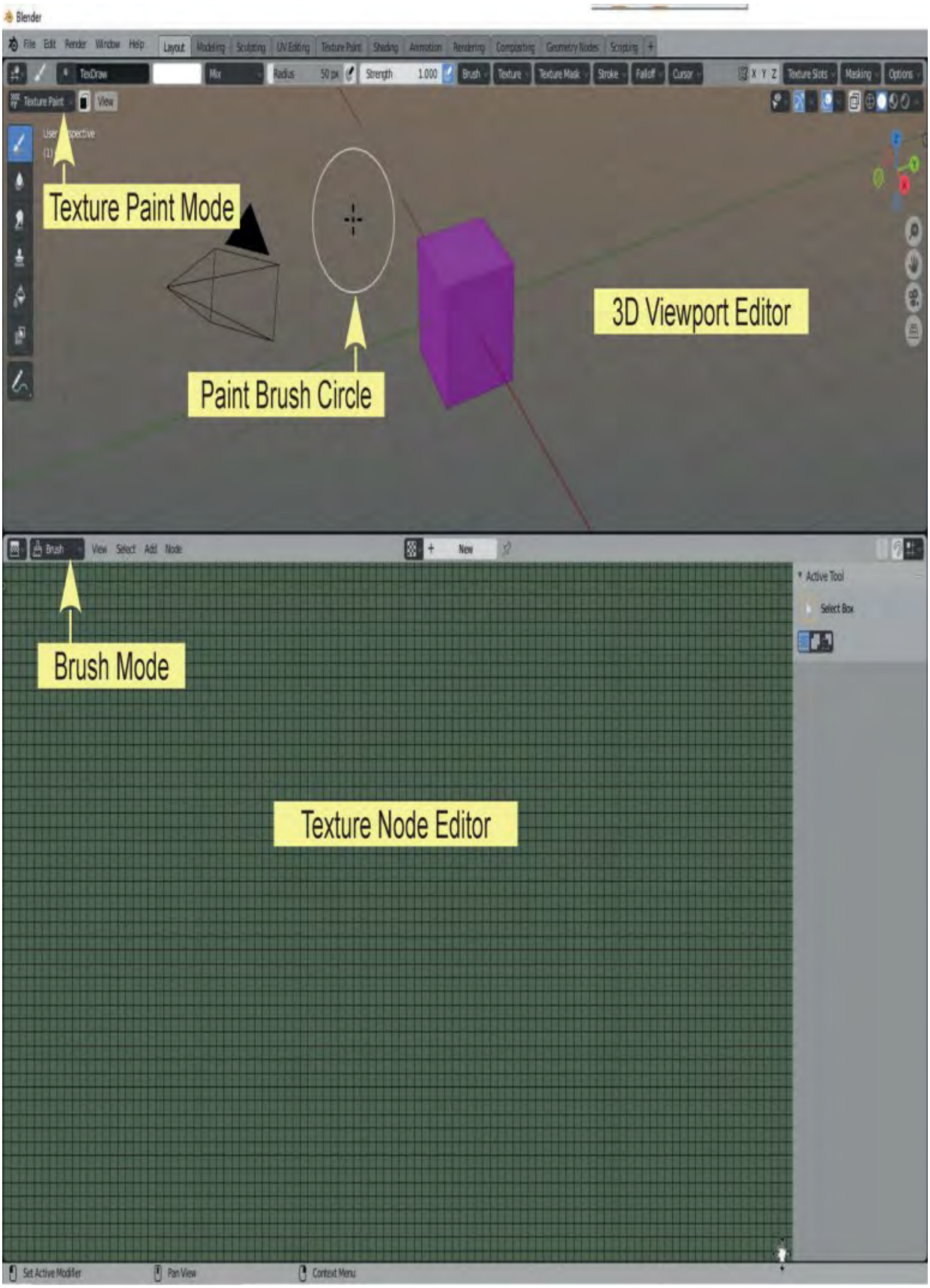

Blender
Texture Paint Mode
Paint Brush Circle
3D Viewport Editor
Brush Mode
Texture Node Editor

Figure 18.2

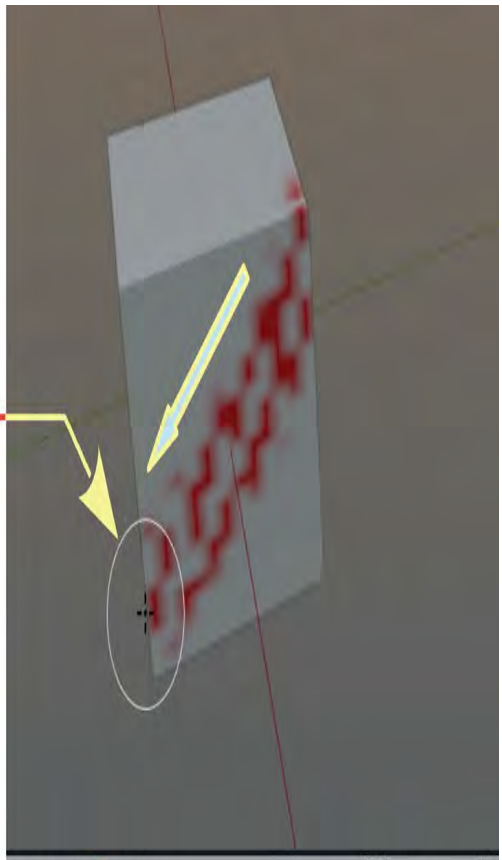
Note: In the default Screen Header you will see that there is a **Texture Paint** Workspace option. For the time being stay with the Workspace you have just created (Figure18.2).

The default Cube Object in the 3D Viewport Editor is displaying purple. Remember; the default Cube has a Material pre-applied. If you enter a New Object it will not have a Material. The purple color indicates that the Cube does not have any Texture applied. As with Material Texture Painting (reference Chapter 17 - 17.8) a Texture has to be painted over an existing Texture.

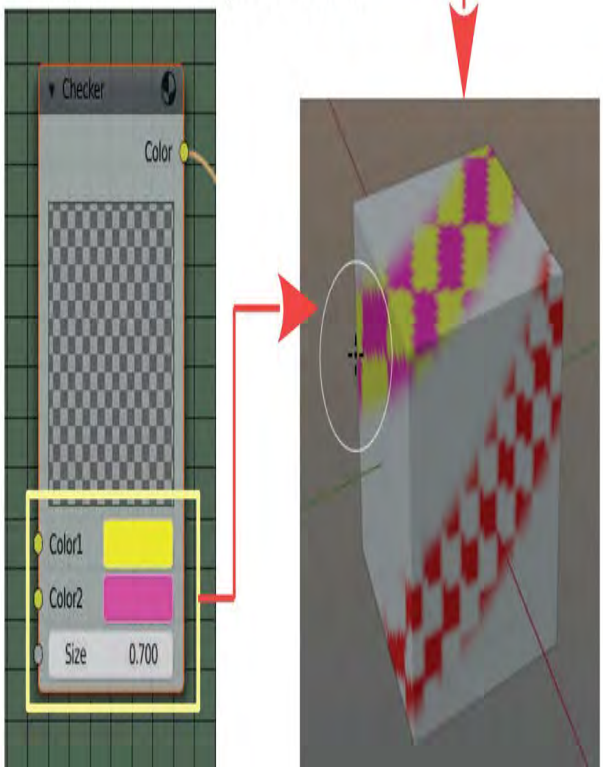
Have the default Cube Object selected in Object Mode in the 3D Viewport Editor and apply a base Texture (Ref. Chapter 17 - 17.8 Texture Painting — Figure 17.41). **Note:** It is not necessary to have the 3D View Editor displaying Object Mode as long as the Cube is selected. With the base Texture applied the Cube in the 3D Viewport Editor in Texture Paint Mode displays gray (the default gray Material color).

In the **Texture Node Editor** in **Brush Mode,** click the **New button** in the center of the Header (New becomes Texture) then check the **Use Nodes** button that displays in the Header. By default you immediately see a **Checker Texture Node** connected to an **Output Node.**

In the 3D Viewport Editor position the **Brush Circle** over the Cube, click, hold and drag the Mouse to paint a Checker Texture on the surface of the Cube.



Adjust values in the Checker Node to alter the Texture drawn in the 3D Viewport Editor.



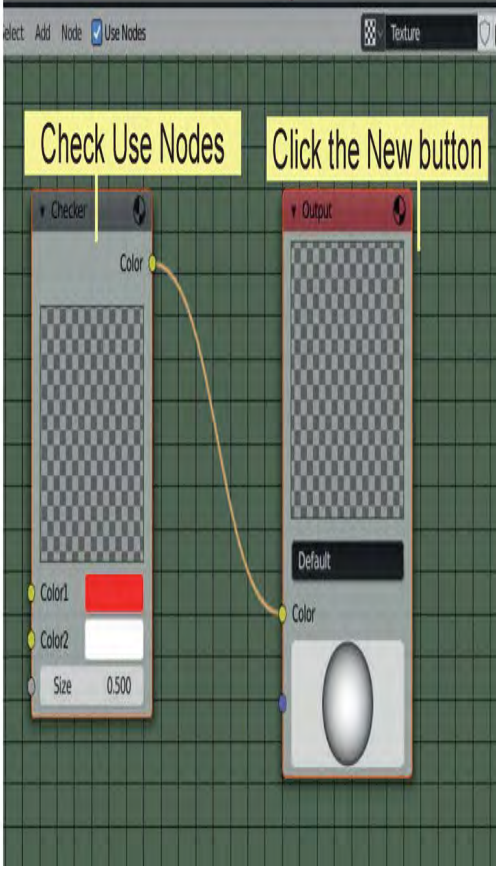
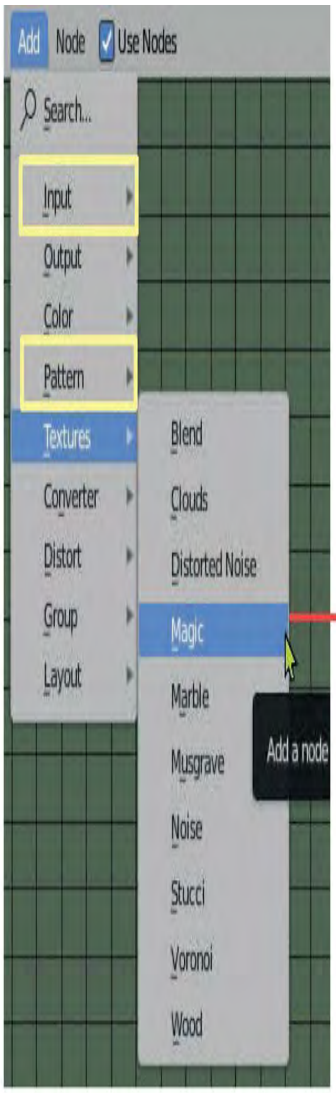
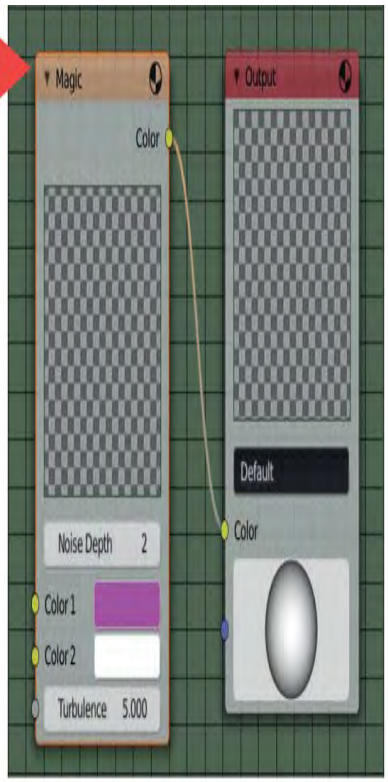
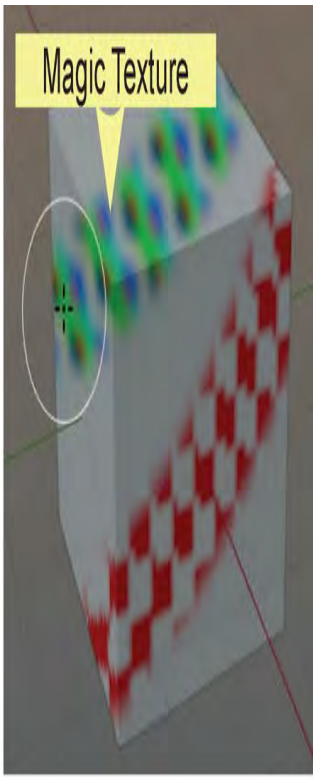


Figure 18.3

Experiment with different Texture Nodes from the **Add menu** in the Texture Node Editor Header.

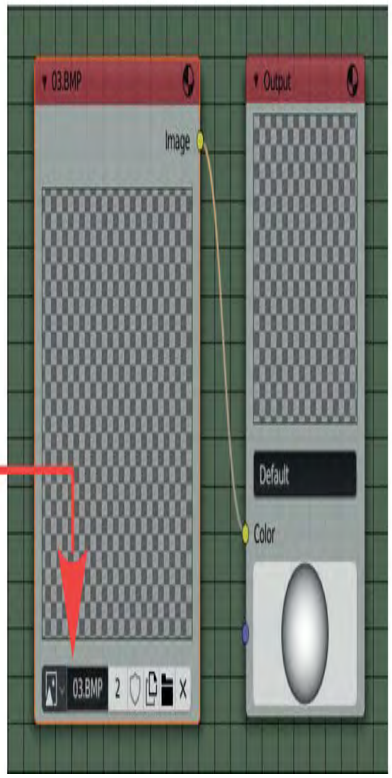






Enter a Image in an **Input – Image Node.**

Try the **Pattern** Options.



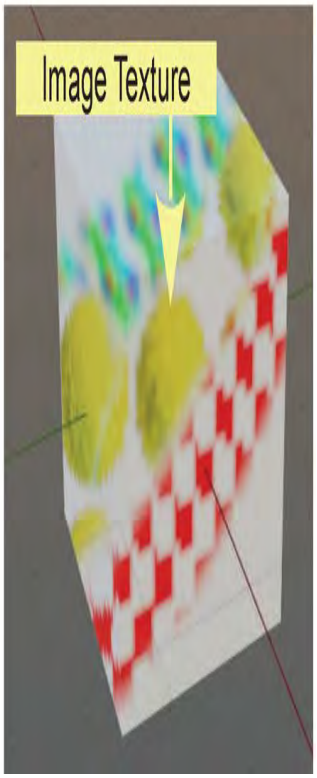


Figure 18.4

Enter a Image in an **Input – Image Node.**

Try the **Pattern** Options.

18.2 The Geometry Node Editor

The **Geometry Node Editor** is accessed from the Editor Type selection menu in an Editor Panel (Figure18.1). At the time of writing Geometry Nodes are under development and will no doubt require a dedicated publication to describe the comprehensive scope of the subject. At this time the Geometry Node Editor will be introduced by demonstrating a simple example using **Blender Version 2.93.5 (Note:** In Blender Version 3.1.0 experimental the Node selection and interface is different).

You may reconfigure the default Blender Screen arrangement creating a Geometry Node Workspace by dragging the upper edge of the Timeline Editor up and changing the Timeline Editor to the Geometry Node Editor. It is suggested, however, that you engage the **Geometry Node Workspace** from the Workspace selection button in the Screen Header. Activating the Workspace from the Header displays the **Spreadsheet Editor,** the **3D Viewport Editor,** the **Geometry Node Editor** with the **Outliner** and **Properties Editors.**

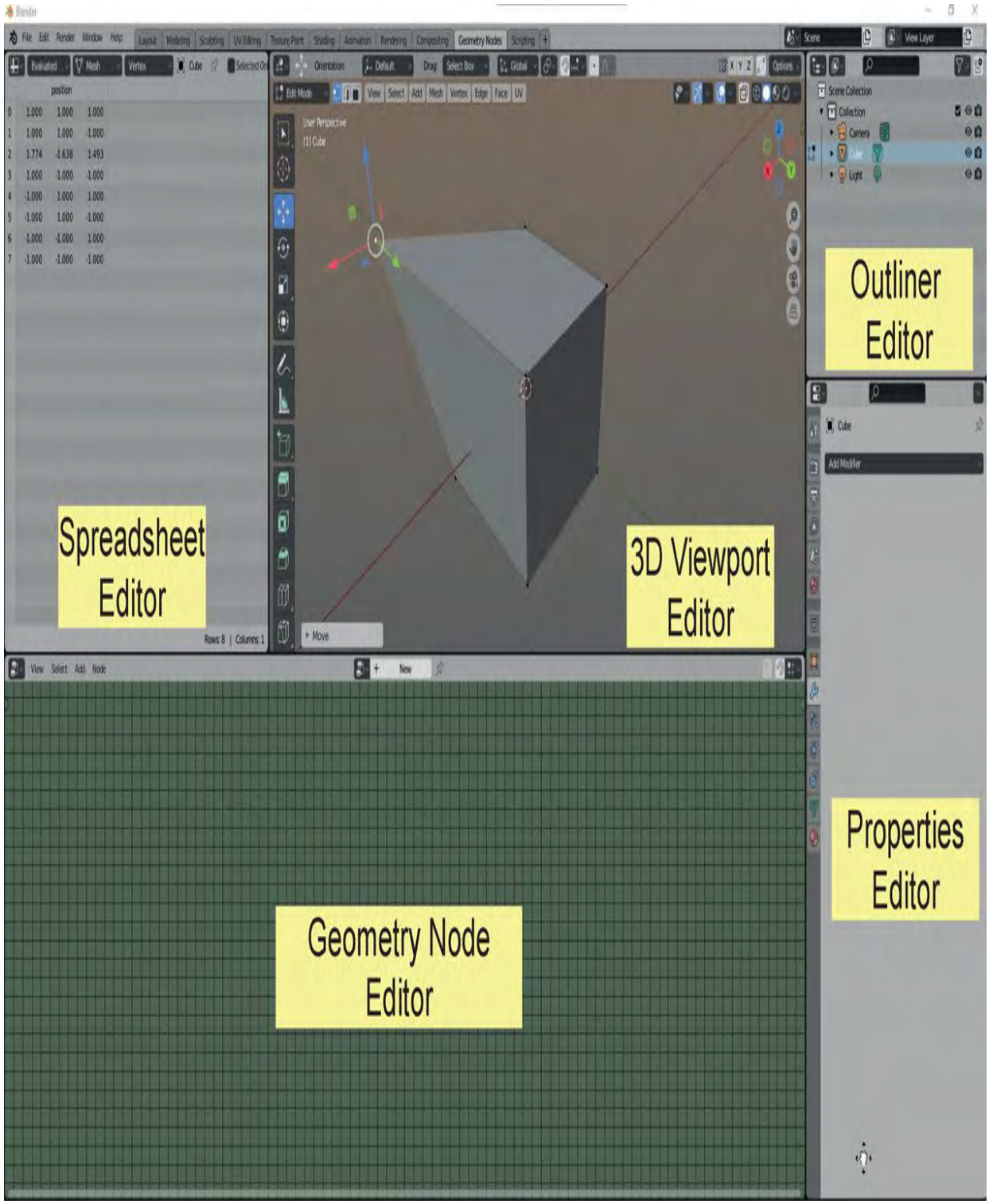

Spreadsheet Editor
3D Viewport Editor
Outliner Editor
Properties Editor
Geometry Node Editor

Figure 18.5

In Figure 18.6 the 3D Viewport has been changed to **Edit Mode** and a single Vertex on the Cube Object has been Translated. In Translating the Vertex you will observe that the relative location of the Vertex is recorded in the **Spreadsheet Editor.**

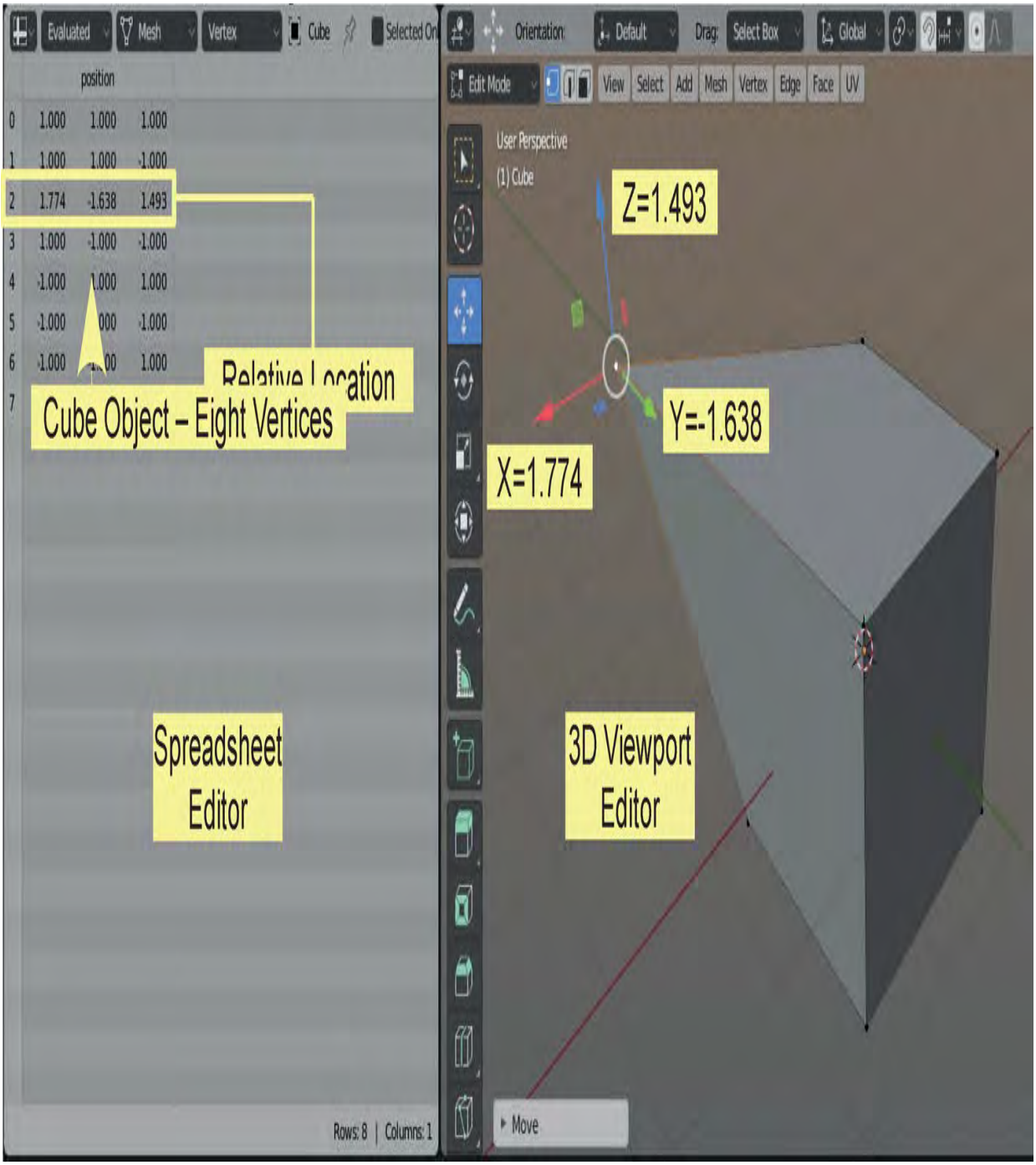


Figure 18.6

With an Object selected in the 3D Viewport Editor, clicking the **New Button** in the Geometry Node Editor Header places a **Group Input Node** connected to a **Group Output Node** in the Editor (Figure 18.7). You may consider these as the beginning and the end of the **Node Pipeline.** Other Nodes entered will be connected between the Input and Output.

The geometry **Node selection menu** is accessed by clicking the **Add Button** in the Header.

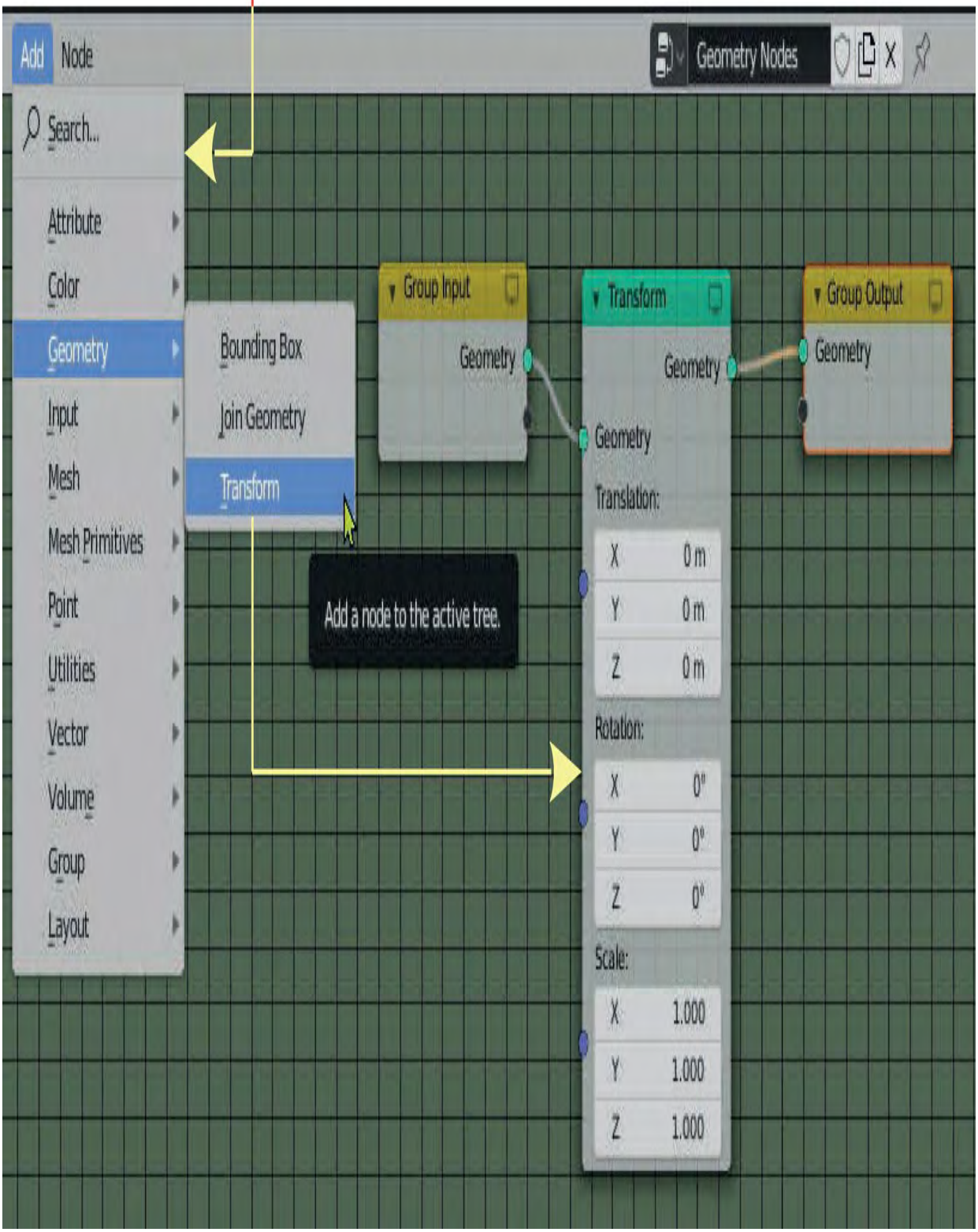


Figure 18.7

Note: At the time of writing Geometry Nodes are a developing component of the Blender program. Additional Nodes are being created and new methods of deployment are being applied. You will, therefore, find that diagrams and instruction provided here for Blender 2.93.5 are not applicable to Blender 3.1.0.

As an introduction a simple vertex displacement demonstration will be employed. In a new Blender Scene delete the default Cube Object in the 3D Viewport Editor and add a Plane. In Edit Mode subdivide the Plane. Number of Cuts: 20 (Figure 18.8).

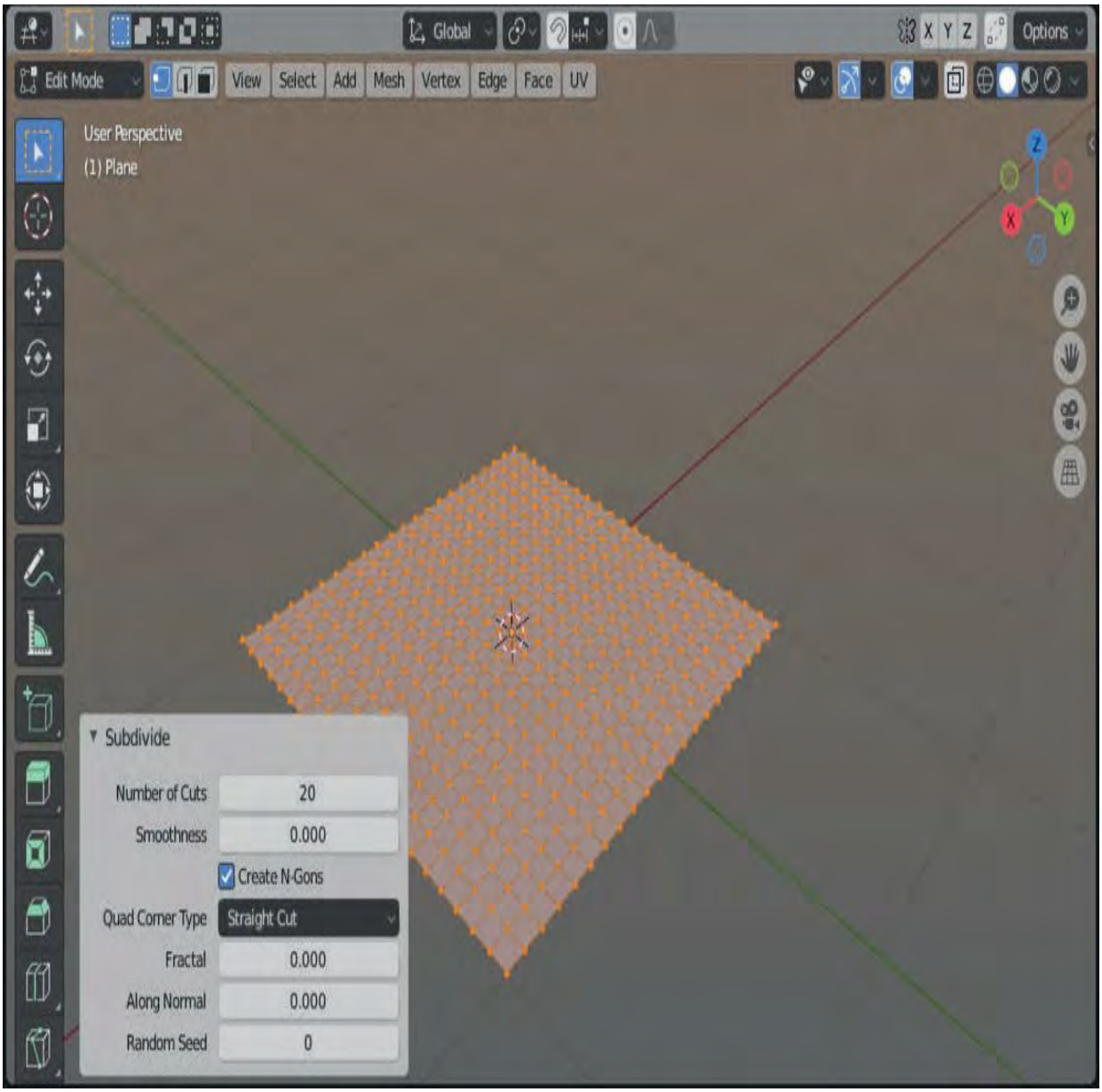


Figure 18.8

Starting the Node Pipeline (Tree)

There are two methods for entering the **Input** and **Output** Nodes in the Geometry Node Editor (**GNE**). You either click The **New button** in the Header or in the Properties Editor, Modifier Properties you select **Geometry Node** in the Modifier Generate category. Clicking the New button in the Geometry Node Editor Header inserts a Geometry Modifier in the **Modifier**

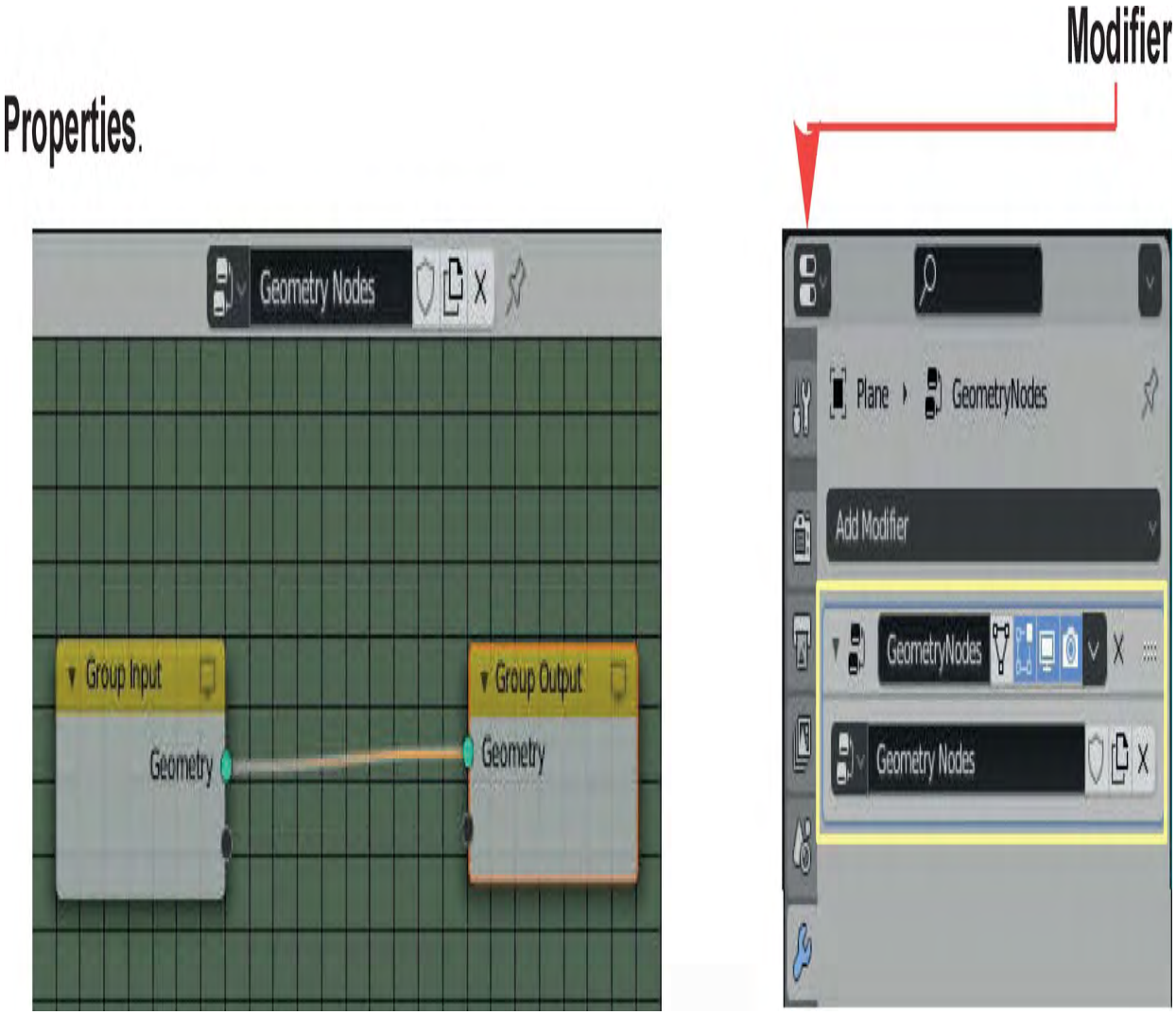
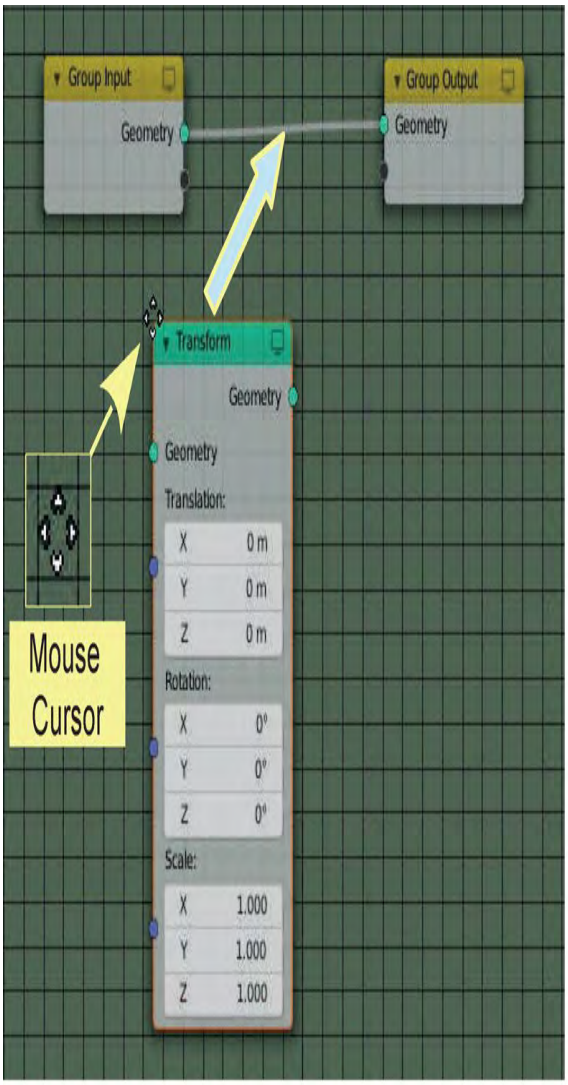


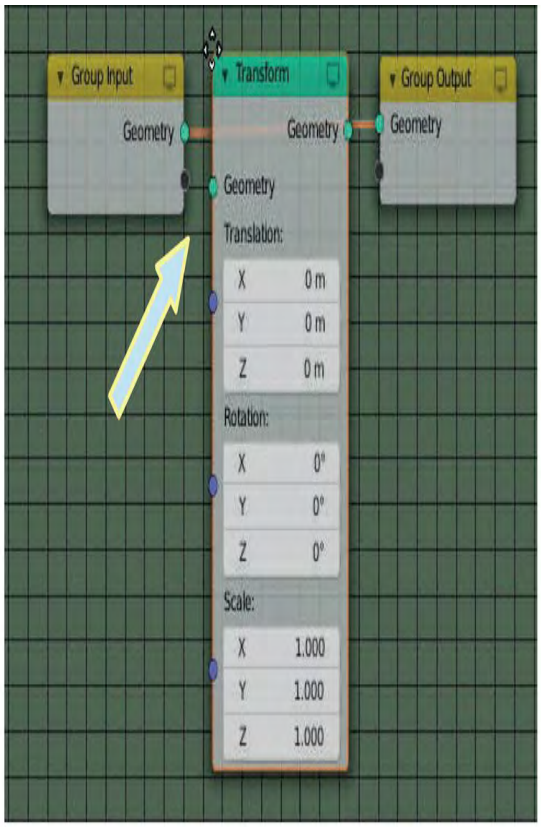
Figure 18.9

Note: The Nodes in the **GNE** are applicable to the Object selected in the 3D Viewport Editor.

In the **GNE** click the **Add button** in the Header or with the **Mouse Cursor** in the **GNE** press **Shift + A Key** for the **Add Node Menu.** Select a **Geometry — Transform Node.**

A **Transform Node** is entered in the GNE in **Grab Mode** as indicated by the change to the Mouse Cursor. Drag the Mouse and position the Transform Node between the Input and Output Nodes. The Transform Node will be automatically connected to the **Geometry Sockets.**


Group Input
Geometry
Group Output
Geometry
Transform
Geometry
Geometry
Translation:
X 0 m
Y 0 m
Z 0 m
Rotation:
X 0°
Y 0°
Z 0°
Scale:
X 1.000
Y 1.000
Z 1.000
Mouse Cursor


Group Input
Geometry
Transform
Geometry
Geometry
Translation:
X 0 m
Y 0 m
Z 0 m
Rotation:
X 0°
Y 0°
Z 0°
Scale:
X 1.000
Y 1.000
Z 1.000
Group Output
Geometry

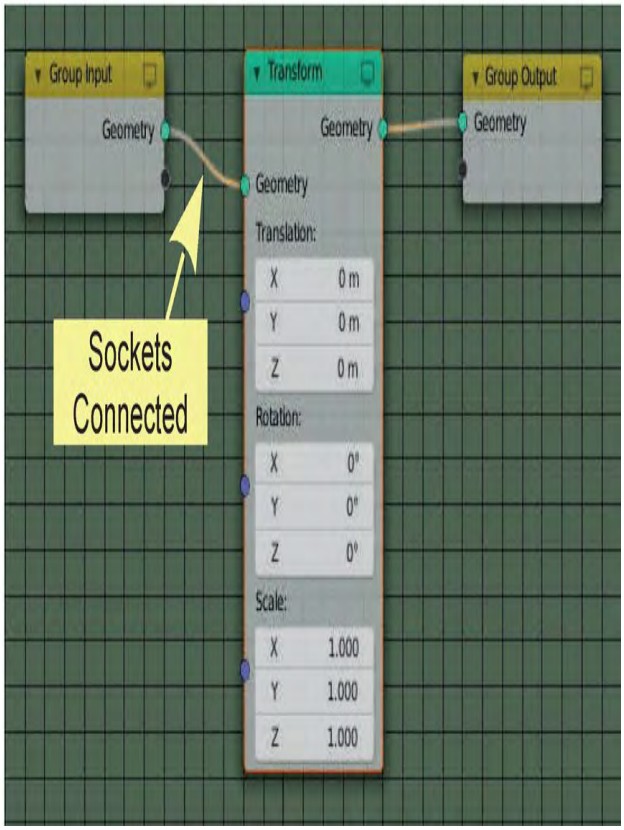

Group Input
Geometry
Transform
Geometry
Geometry
Translation:
X 0 m
Y 0 m
Z 0 m
Rotation:
X 0°
Y 0°
Z 0°
Scale:
X 1.000
Y 1.000
Z 1.000
Group Output
Geometry
Sockets Connected

Figure 18.10

Note: The Nodes are applicable to the Plane Object which is selected in the 3D Viewport Editor. The Plane has been subdivided in Edit Mode, therefore, the **Spreadsheet Editor** displays the coordinates for each Vertex of the subdivided Plane.

In the Transform Node increase the Translation Z Axis value to 0.9 m. By placing the 3D Viewport Editor in Edit Mode and having all Vertices selected (Press the A Key) you will see the Plane Translated 0.9 m above the Mid Plane Grid. The vertices remain in situ.

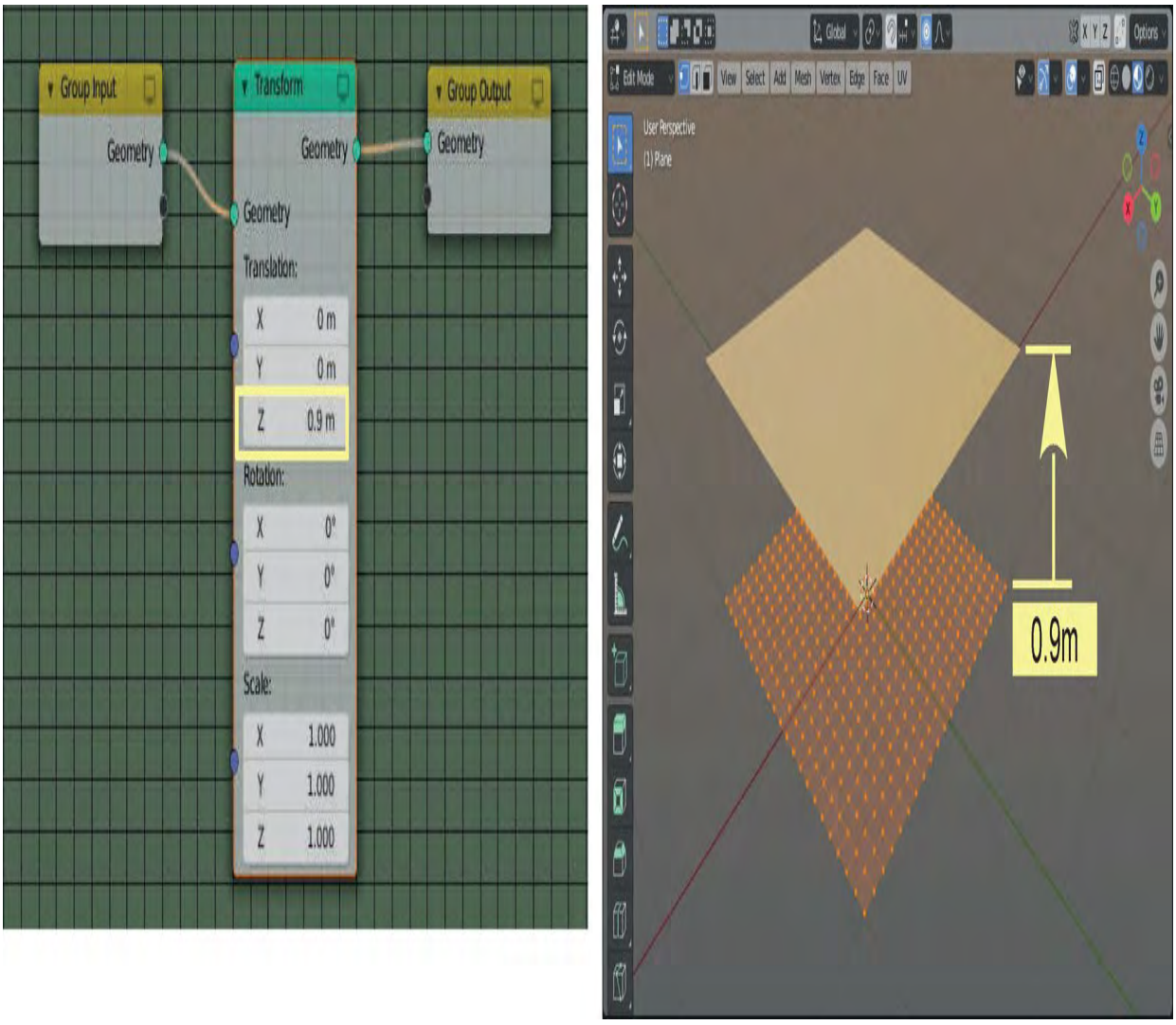


Figure 18.11

The foregoing has demonstrated how values set in a geometry Node in the GNE affect an Object in the 3D Viewport Editor. To develop the concept perform the following exercise.

In a new Blender Scene replace the default Cube with an **Icosphere.** The Icosphere is a little more interesting while having a relatively low number of Vertices. There are some fantastic demonstrations of Geometry Node applications on the internet but be aware that with high vertex count comes a demand on computer power.

Scale the **Icosphere** way down and park it to one side in the Viewport. Add a nice bright Material to the Icosphere.

Add a **Plane Object** to the Scene and in **Edit Mode,** Subdivide X 1 creating 9 Vertices and Scale up X 4.

Have the **Geometry Node Workspace** open and with the Plane selected click **New** in the GNE Header creating **Group Input** and **Output Nodes.**

In the GNE Header click the **Add button** and select a **Point — Instance Node.**

An **Instance** of an Object is a Copy of the Object. In this particular case, **Point** refers to a Vertex. The Point Instance Node creates a copy of an Object and places a copy at the position of each Vertex of an Object entered in the Node.

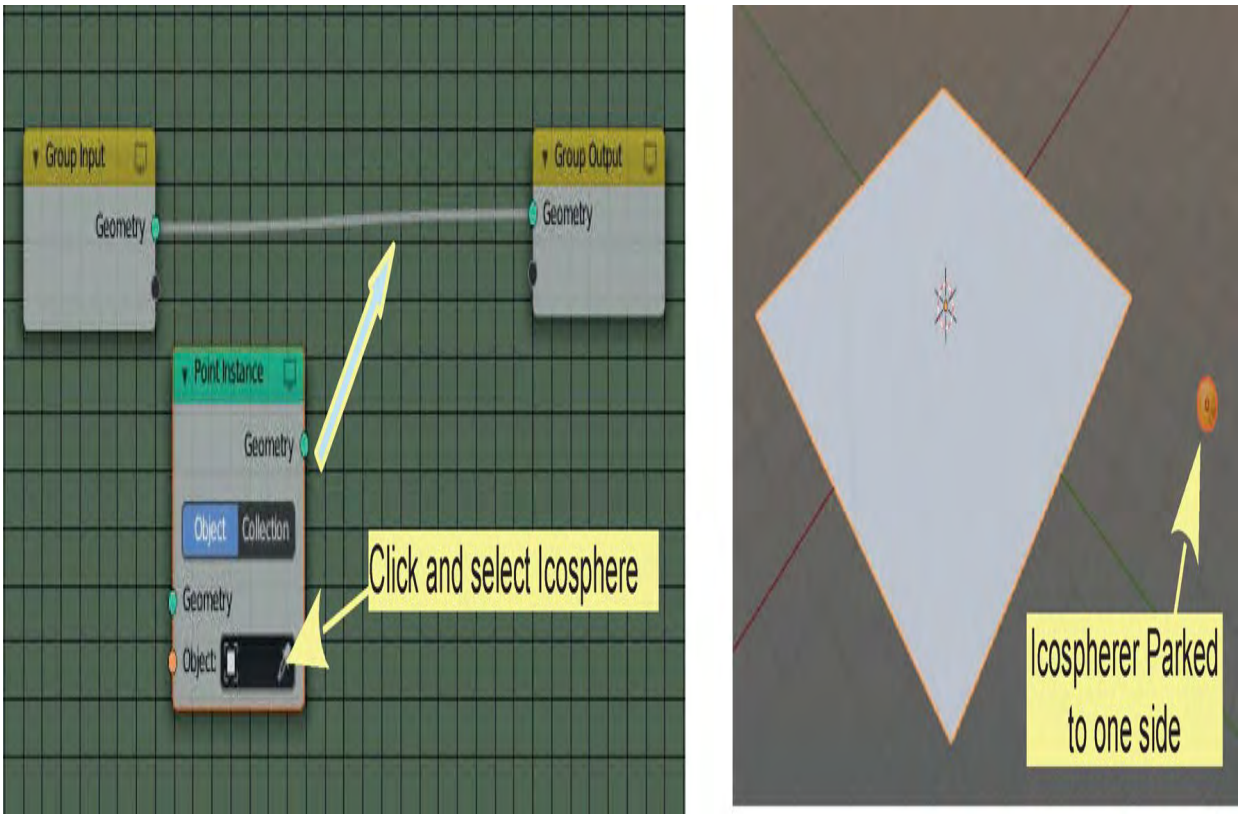


Figure 18.12

Position the Point — Instance Node between the Input and Output Nodes.

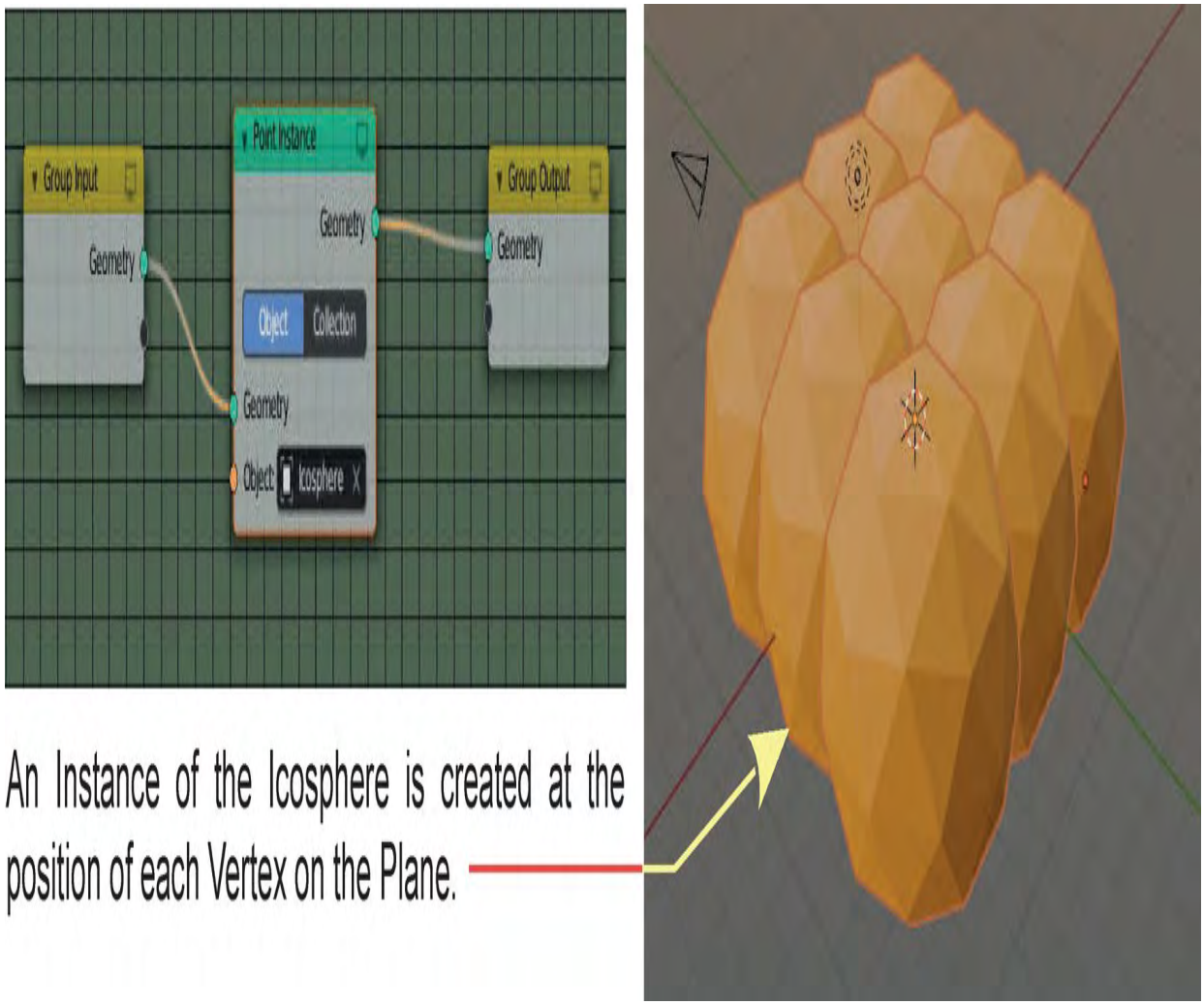


Figure 18.13

Insert a **Point — Scale Node** entering X: 0.300, Y: 0.300, Z: 1.000

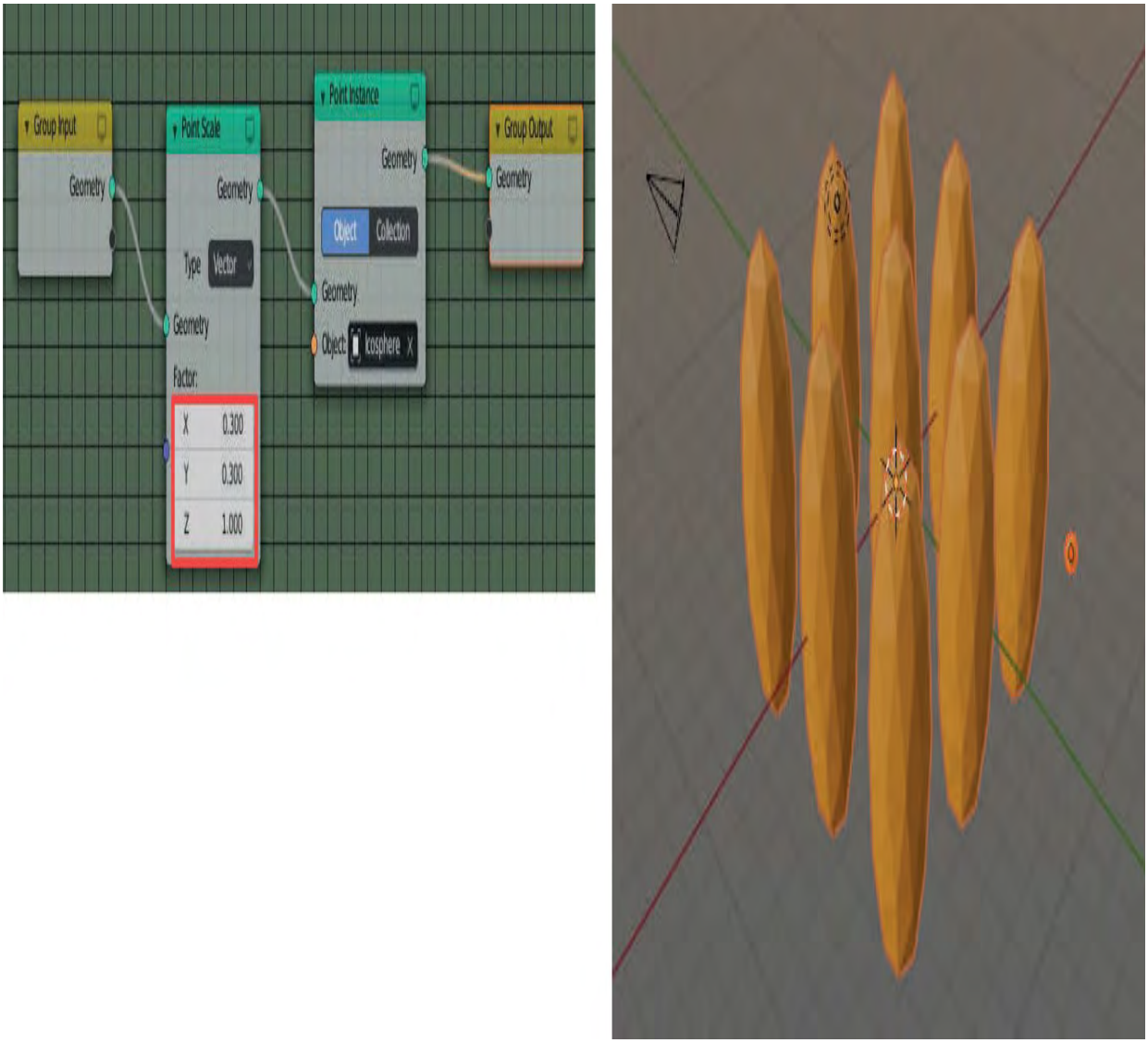
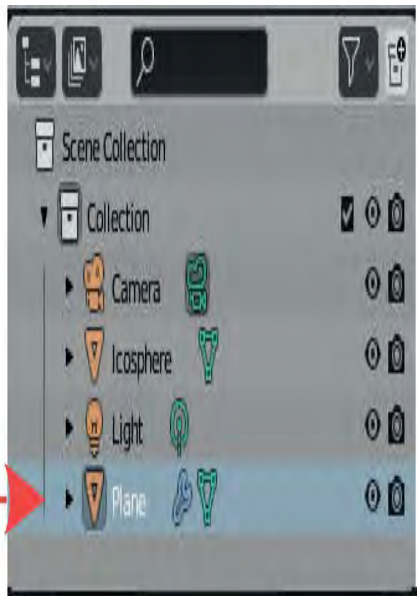


Figure 18.14

With the **Point — Scale Node** placed before the Instance Node in the Pipeline, the **Factor** values in the Node control the Scale of the Instance Objects.

Note: To see Material color of the Instance Objects have the 3D Viewport Editor in **Material Preview Mode.**

At this point the original Plane Object is not visible in the 3D Viewport Editor. To ensure that the Plane is selected make sure it is selected by clicking **Plane** in the **Outliner Editor**.



Move the Subdivision Surface Modifier to the top of the stack.

Figure 18.15

With the Plane selected, in the Properties Editor, Modifier Properties add a **Subdivision Surface Modifier.** The Modifier is placed in the Modifier Stack below the Geometry Node Modifier. With the default Modifier values the surface of the Instance Objects are refined but you will no longer see Material Color.

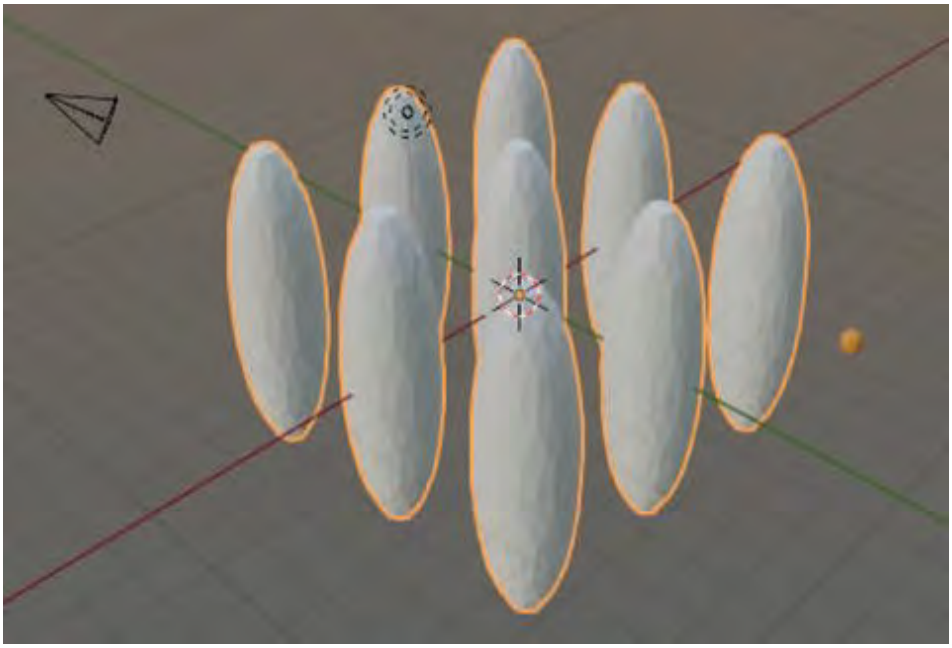


Figure 18.16

When the Subdivision Surface Modifier is at the top of the Stack the default Level and Render values increase the number of Instances in the 3D Viewport Editor. Make Level Viewport: 0.

Change the 3D Viewport Editor to Edit Mode, Wireframe Display Mode. Select a single Vertex. Press the G Key (grab), increase the Circle of Influence to encapsulate other Vertices then Translate the selected Vertex to deform the Instance display.

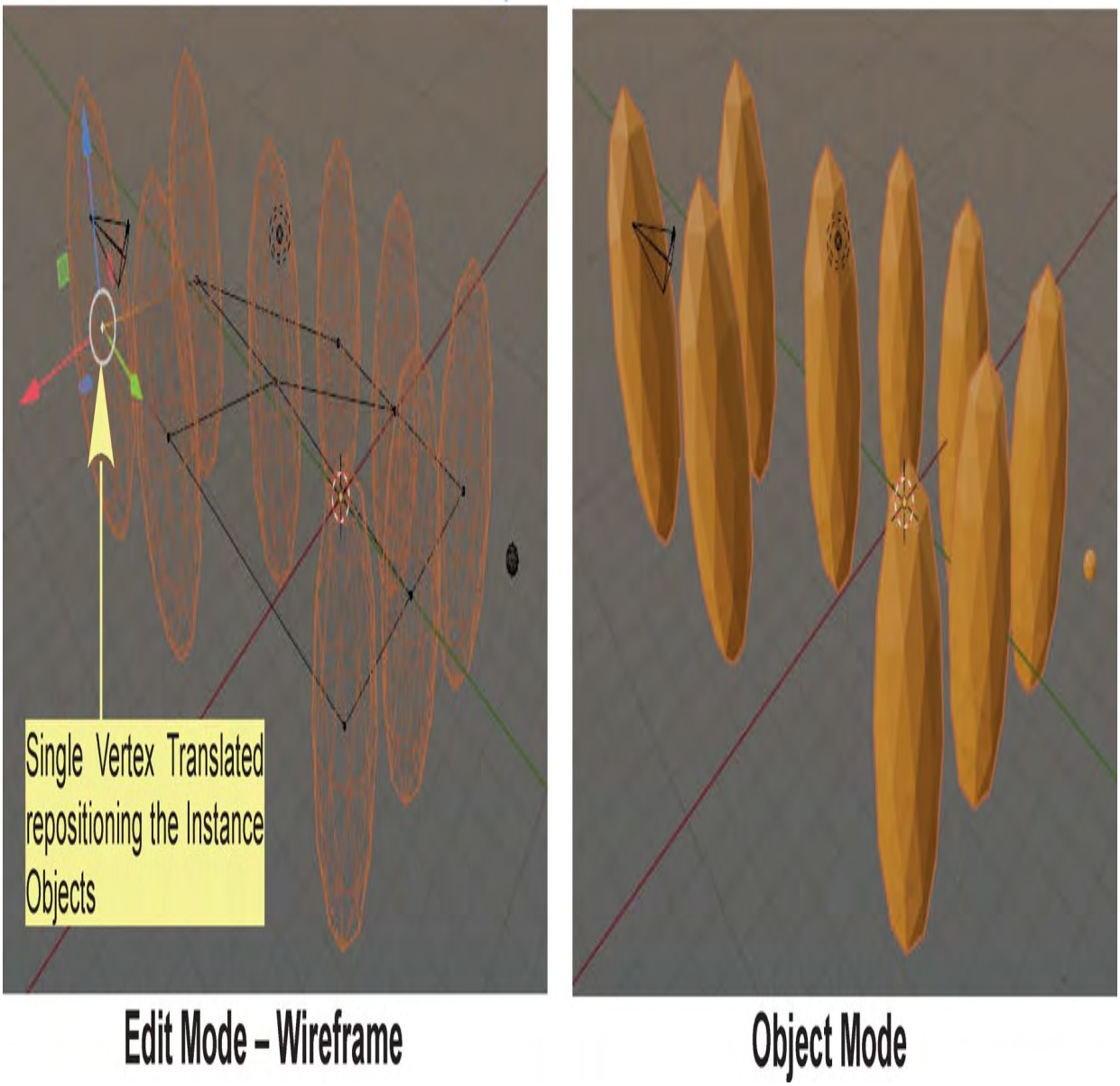
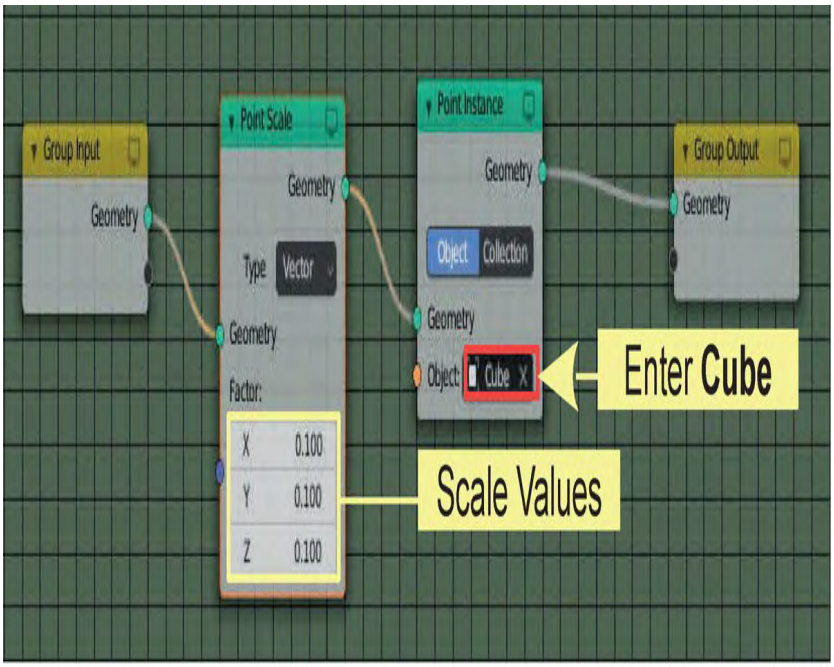


Figure 18.17

This simple procedure demonstrates that a display in the 3D Viewport Editor is controlled by Geometry Nodes in conjunction with Modifiers in the Properties Editor.

Add a Cube Object to the Scene. Apply the Cube as an Instance Object to the Icosphere. This is applying an Instance to an Instance.

Geometry Nodes applied to the **Icosphere** used previously as the Instance Object


Group Input
Geometry
Point Scale
Geometry
Type
Vector
Geometry
Factor:
X
0.100
Y
0.100
Z
0.100
Point Instance
Geometry
Object
Collection
Geometry
Object:
Cube
Group Output
Geometry
Enter Cube
Scale Values

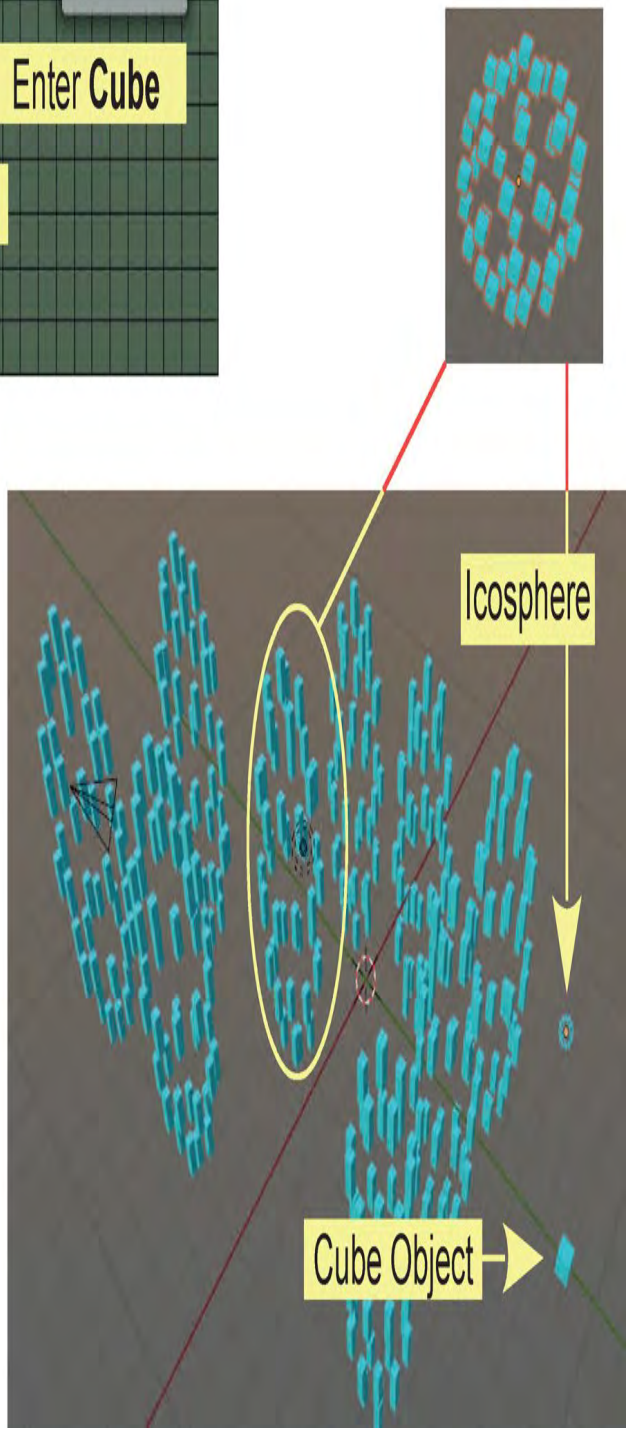

Icosphere
Cube Object

Figure 18.18

Remember: Geometry Nodes are under development. The intention in this section is merely to make you aware of their existence and introduce their capabilities.

18.3 Transparency

To demonstrate Transparency a Plane Object will be turned into a pane of glass. Add a Plane to the default Blender Scene. Stand the Plane on edge and position in front of the default Cube. Give the Cube a Material with Nodes active (Figure 18.19).

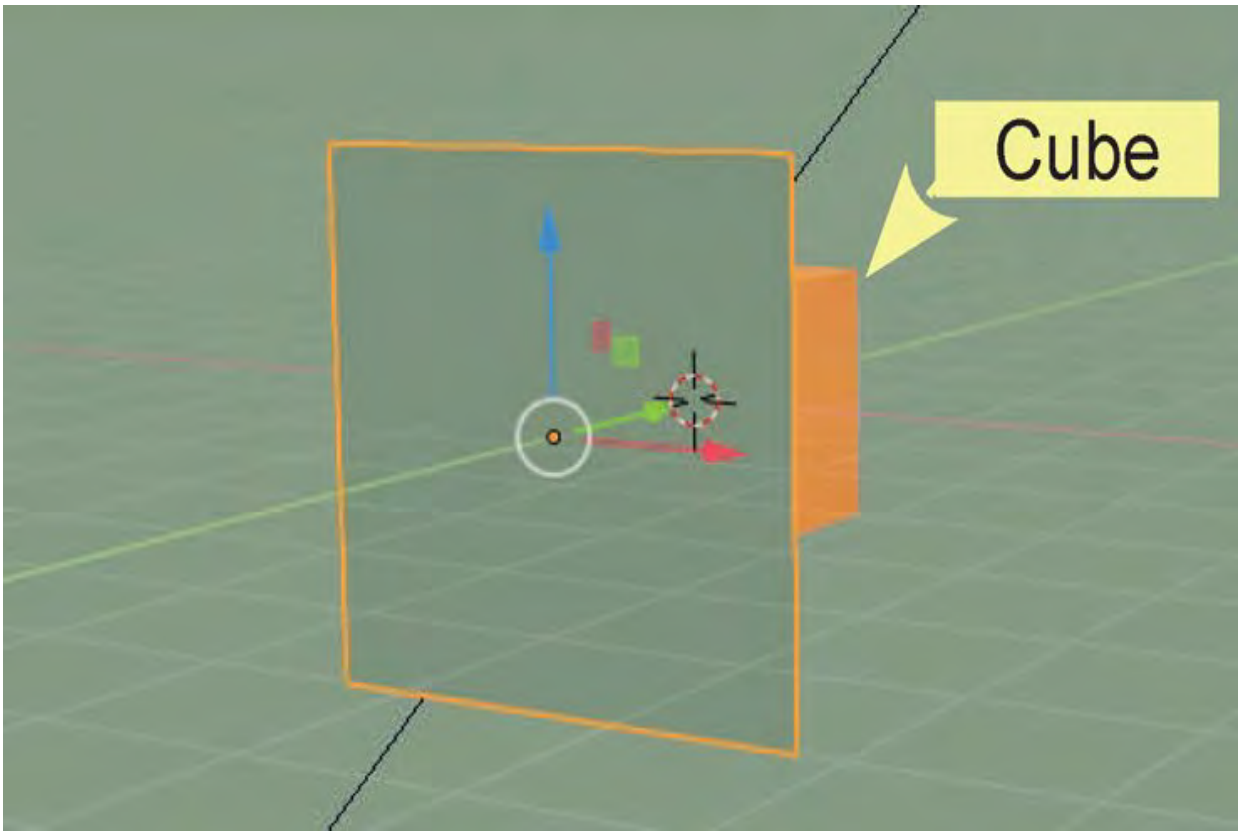


Figure 18.19

Change the default Point Lamp to an Area or Area Light.

Ensure that the **3D Viewport Editor** is in **Rendered Viewport Display Mode.**

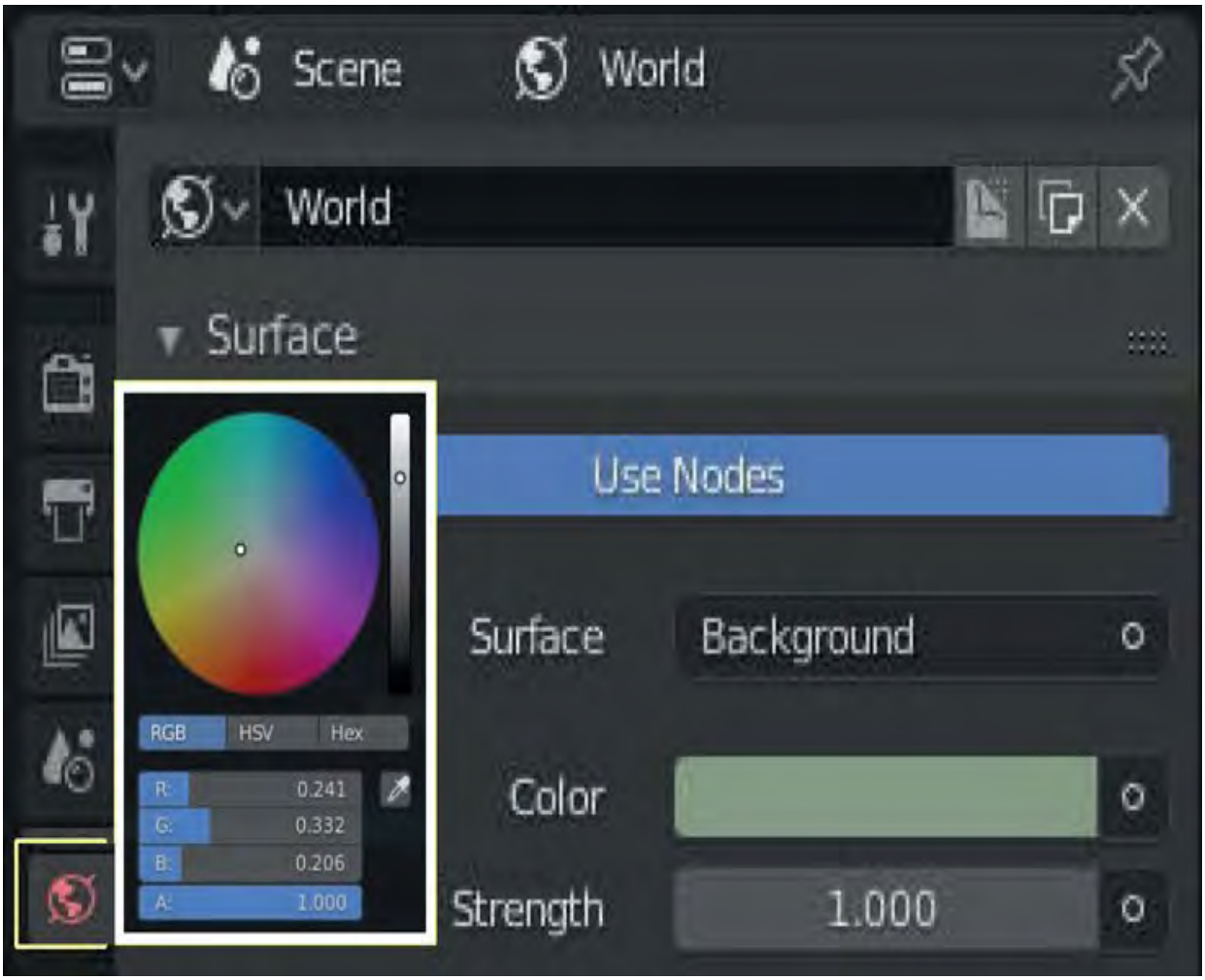


Figure 18.20

In the **Properties Editor, World Properties, Surface Tab** set the Background as shown in Figure 18.20.

Select the Plane Object, add a Material, then in The Shader Editor create the Node arrangement in Figure 18.21. Pay attention to the settings.

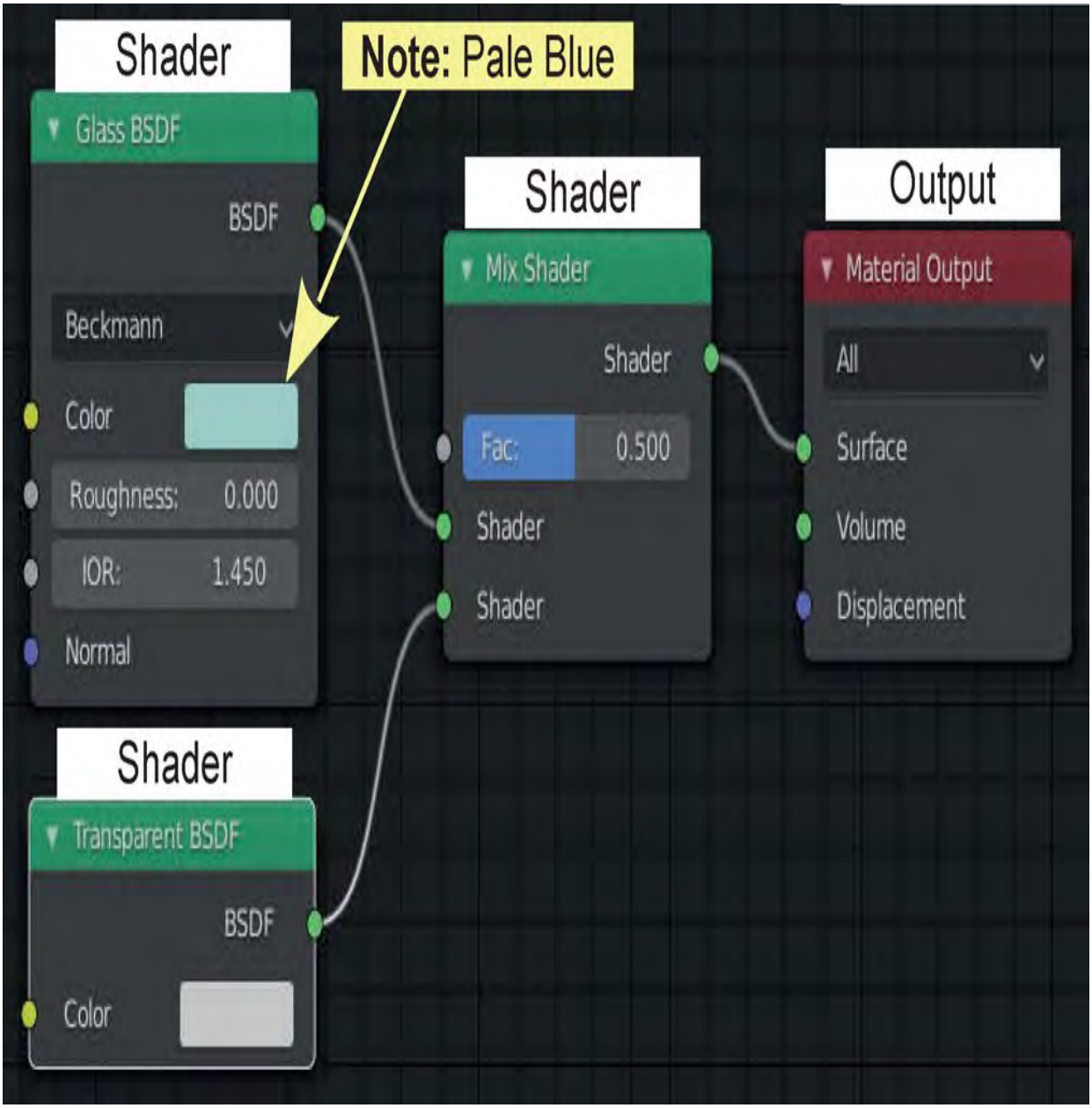

Shader
Note: Pale Blue
Glass BSDF
BSDF
Beckmann
Color
Roughness: 0.000
IOR: 1.450
Normal
Shader
Mix Shader
Shader
Fac: 0.500
Shader
Shader
Output
Material Output
All
Surface
Volume
Displacement
Shader
Transparent BSDF
BSDF
Color

Figure 18.21

In the Material Properties, Settings Tab change the Blend Mode to **Alpha Blend**.

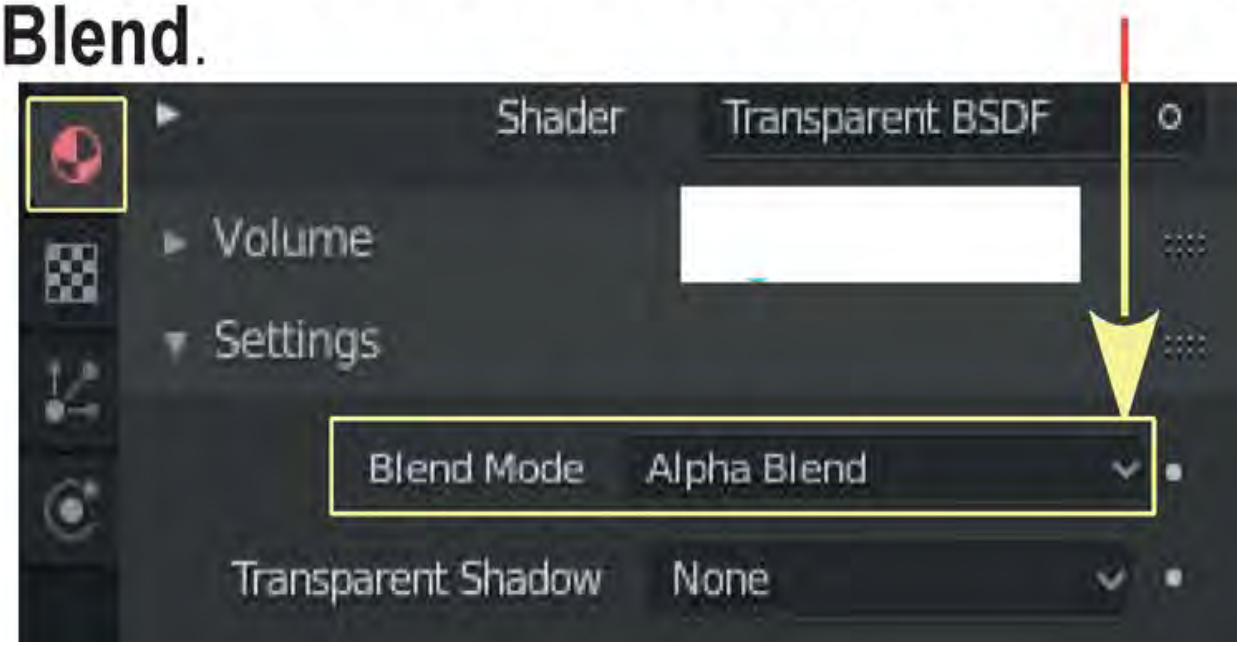


Figure 18.22

In the Render Properties, check **Screen Space Reflections** and **Refraction**.

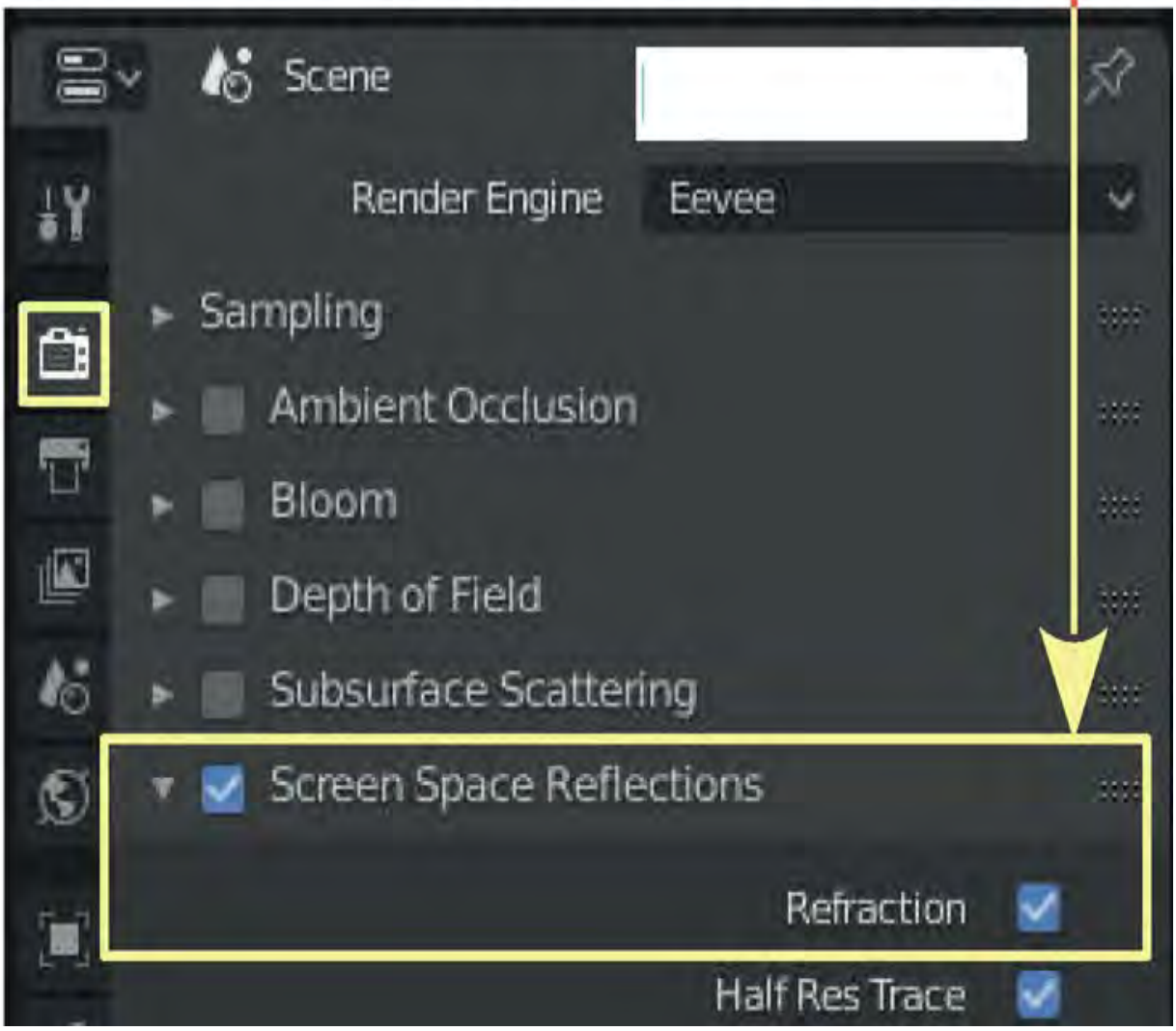


Figure 18.23

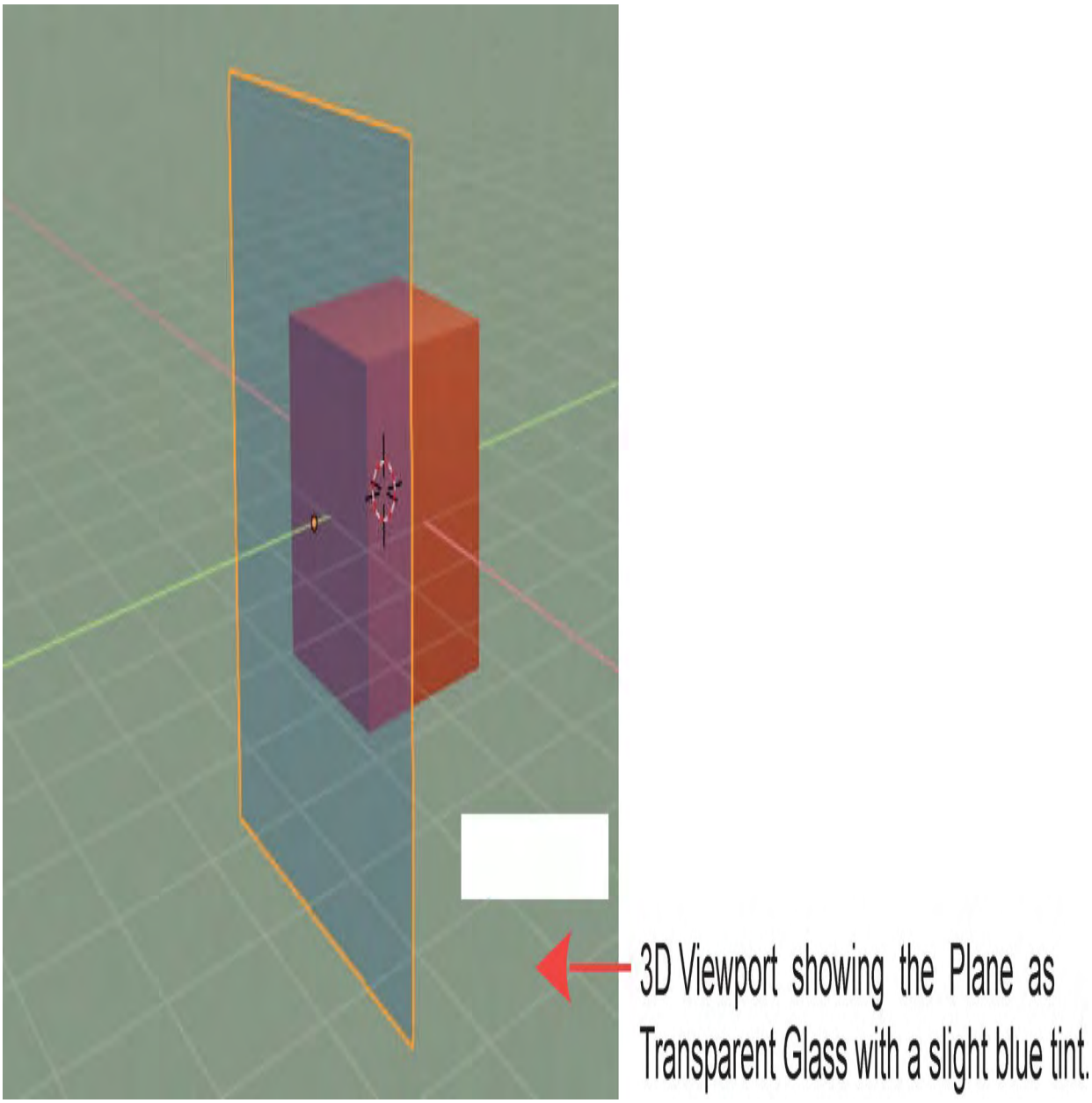


Figure 18.24

With the identical Node arrangement but with an HDRI image (Ref Chapter 15 — 15.9) used in the World Background fantastic results can be generated.

Figure 18.25 shows the pane of glass partially in front of the Cube. With the view rotated the glass reflects light cast by the background HDRI image.

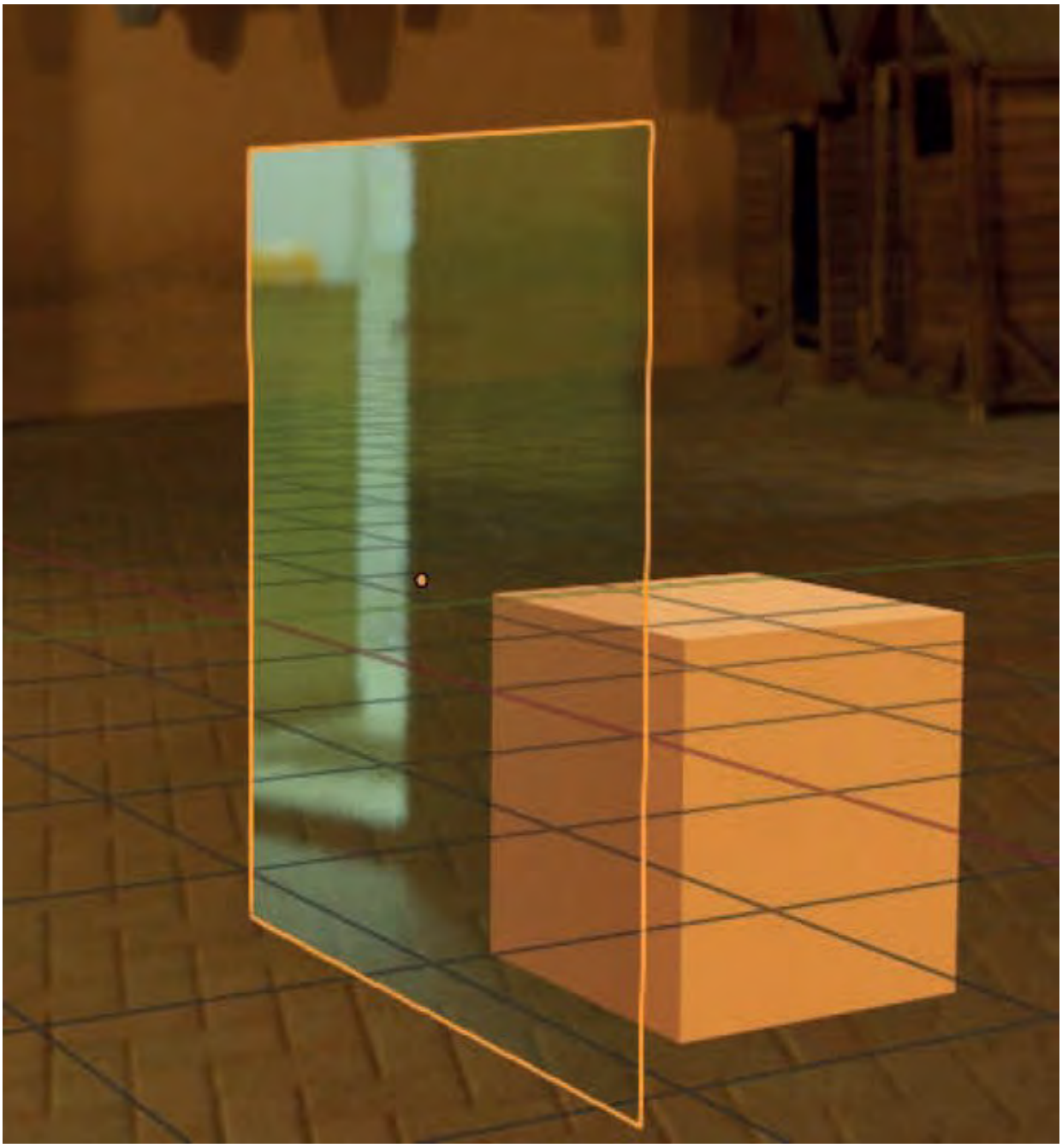


Figure 18.25

18.4 Volumetric Lightinc

Volumetric Lighting is a technique used in 3D computer graphics to add lighting effects to a rendered scene. It allows the viewer to see beams of light

shining through the environment; seeing sunbeams streaming through an open window is an example of Volumetric Lighting, also known as God rays.

Volumetric Lighting requires that a cubic volume of space is defined in which the lighting effect will take place. Objects are encapsulated by the volume and, therefore, affected by the light and the camera in the Scene is positioned to capture the effect.

To demonstrate, the Scene arrangement shown in Chapter 15 - Figure 15.5 will be used with identical Materials applied to the Objects. The Camera is positioned to capture the three Objects in the View set above the Plane (Figure 18.26).

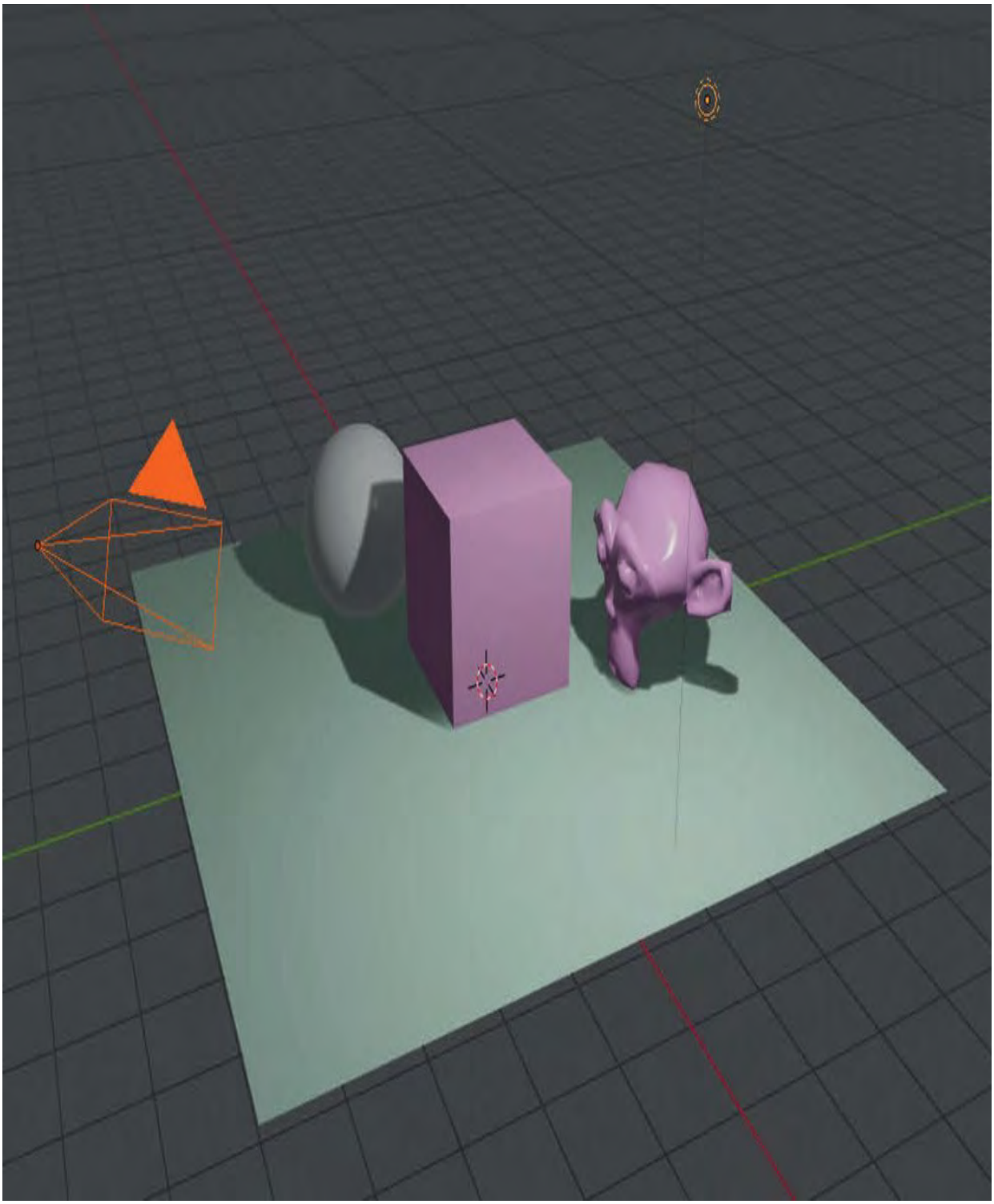
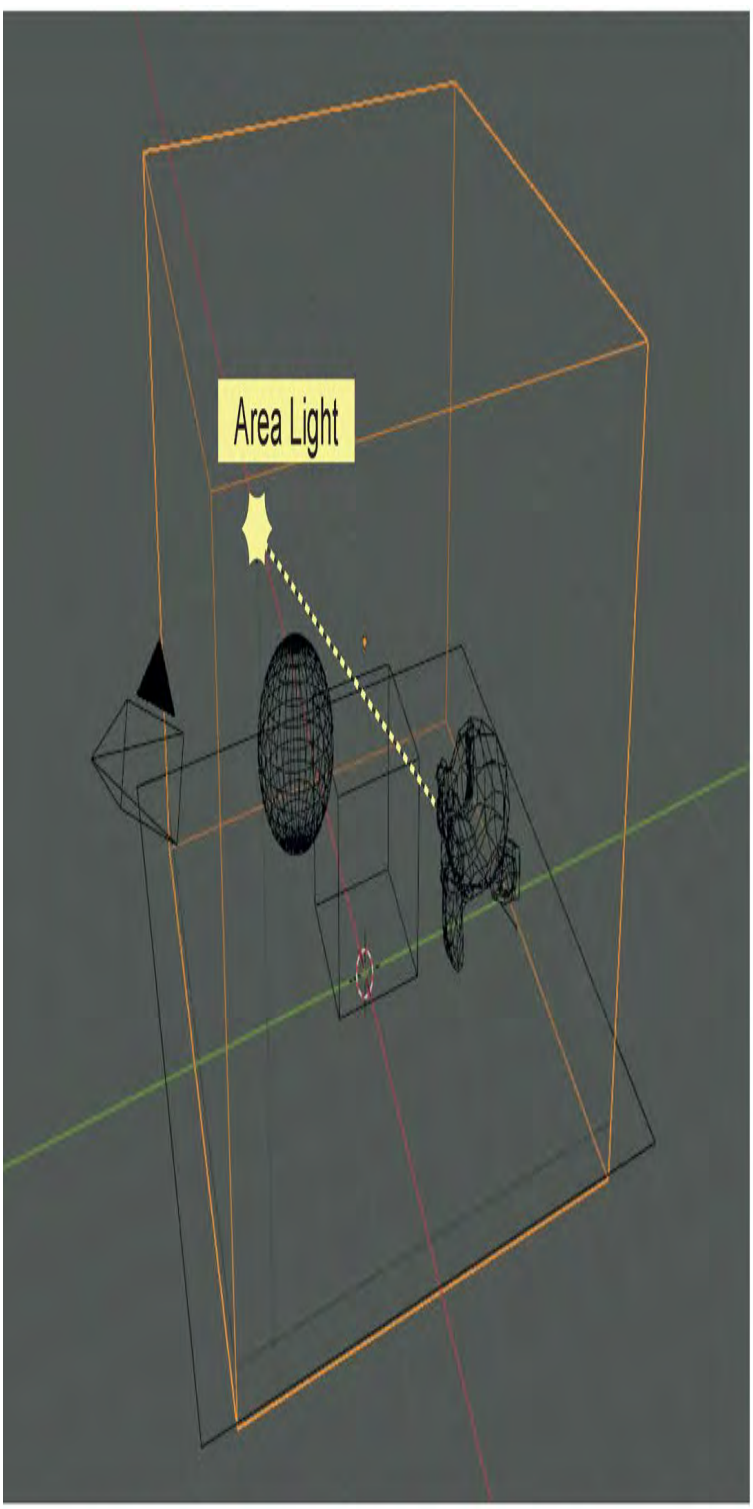


Figure 18.26

The one exception to this Scene Arrangement will be the positioning of the Light Object.

In Wireframe Display Mode add a Cube Object, Scale and position as shown in Figure 18.27. The bottom of the new Cube is just below the Plane. The Light in the Scene is an Area Light with a pale green color positioned and directed at the Objects.


Area Light

Light
Light
Light
Preview
Light
Point
Sun
Spot
Area
Color
Power
1000 W
Diffuse
1.00
Specular
1.00
Volume
1.00
Shape
Rectangle
Size X
0.1 m
Y
0.1 m
Custom Distance

Figure 18.27

Select the **new Cube** in the 3D Viewport Editor. In the Properties Editor, Material Properties, add a Material with Nodes active with the default settings. Divide the 3D Viewport Editor making one half the Shader Editor. Delete the Principled BSDF Node and replace it with a Principled Volume Node (Figure 18.28). Note: In the **Properties Editor, Render Properties, Volumetrics** are active by default.

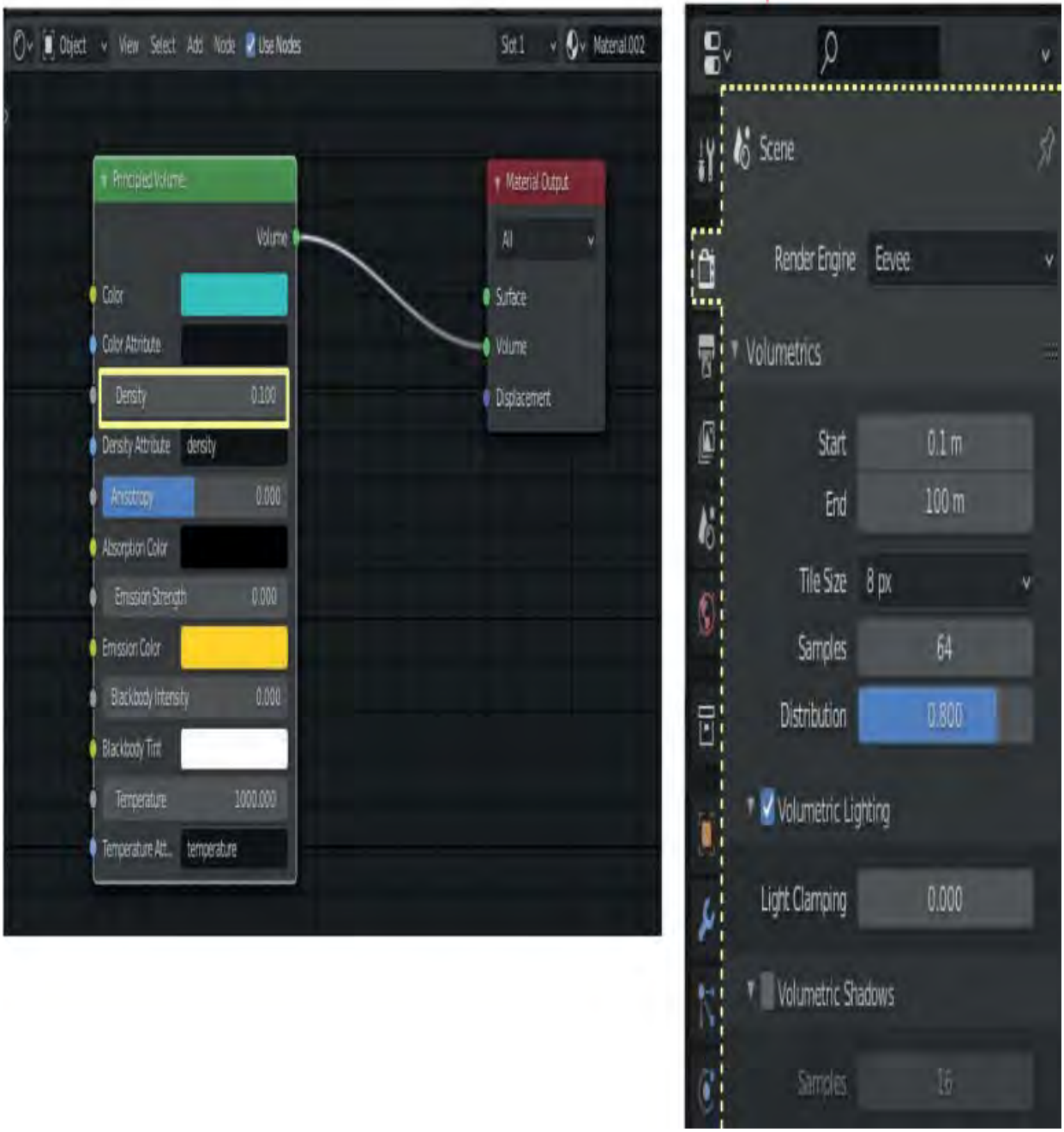


Figure 18.28

In the **Principled Volume Node** set values as shown taking particular attention to the **density value.**

With the 3D Viewport Editor in **Rendered Viewport Shading Mode** you see the Light dispersed through the volume defined by the new Cube. Camera View shows the Objects in the Scene bathed in the pale misty light (Figures 18.29 and 18.30).

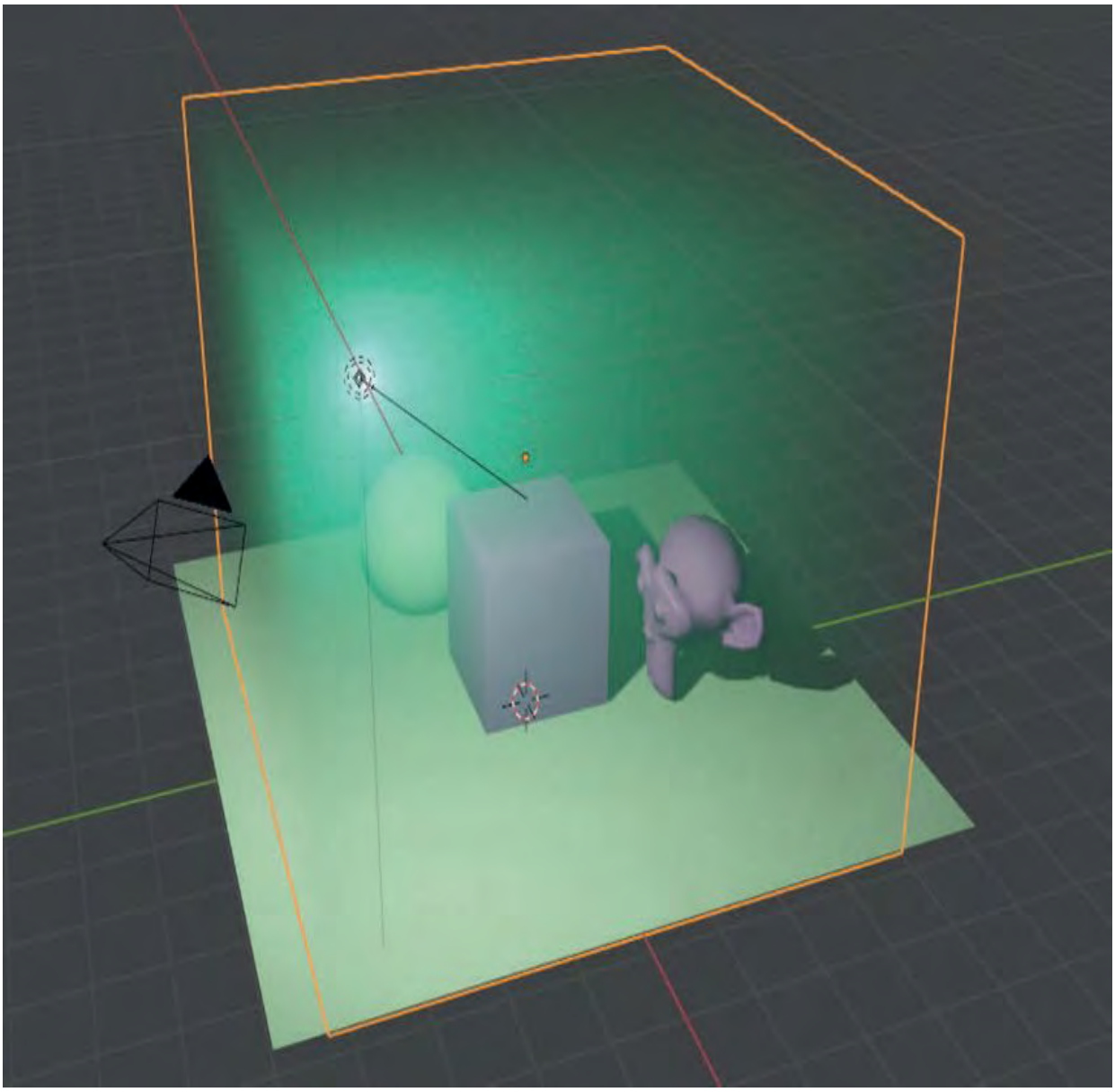


Figure 18.29

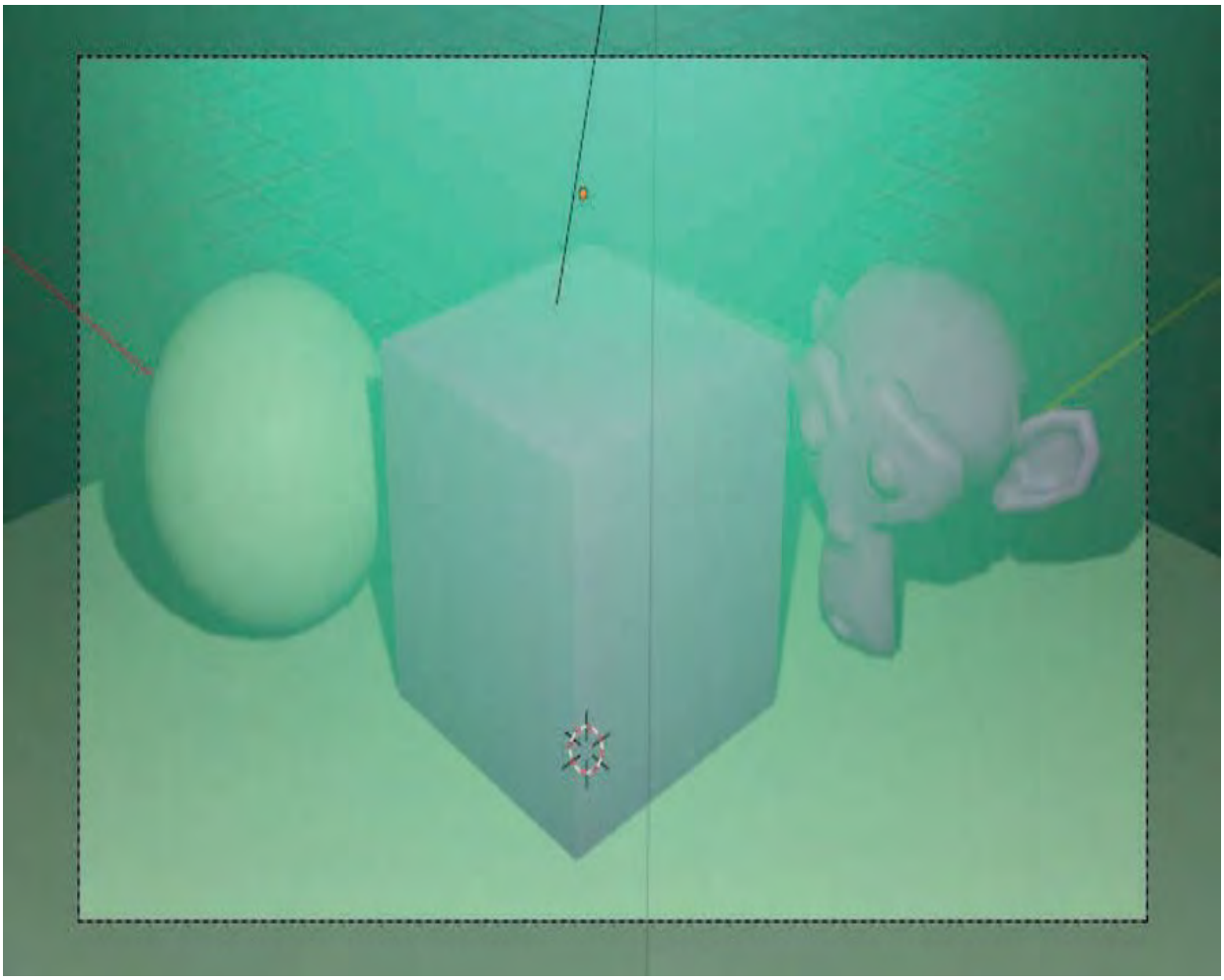


Figure 18.30

By having Objects inside a room with a window with the Light outside. The Camera inside the the room and everything encapsulated by the Light Volume you see light streaming in.

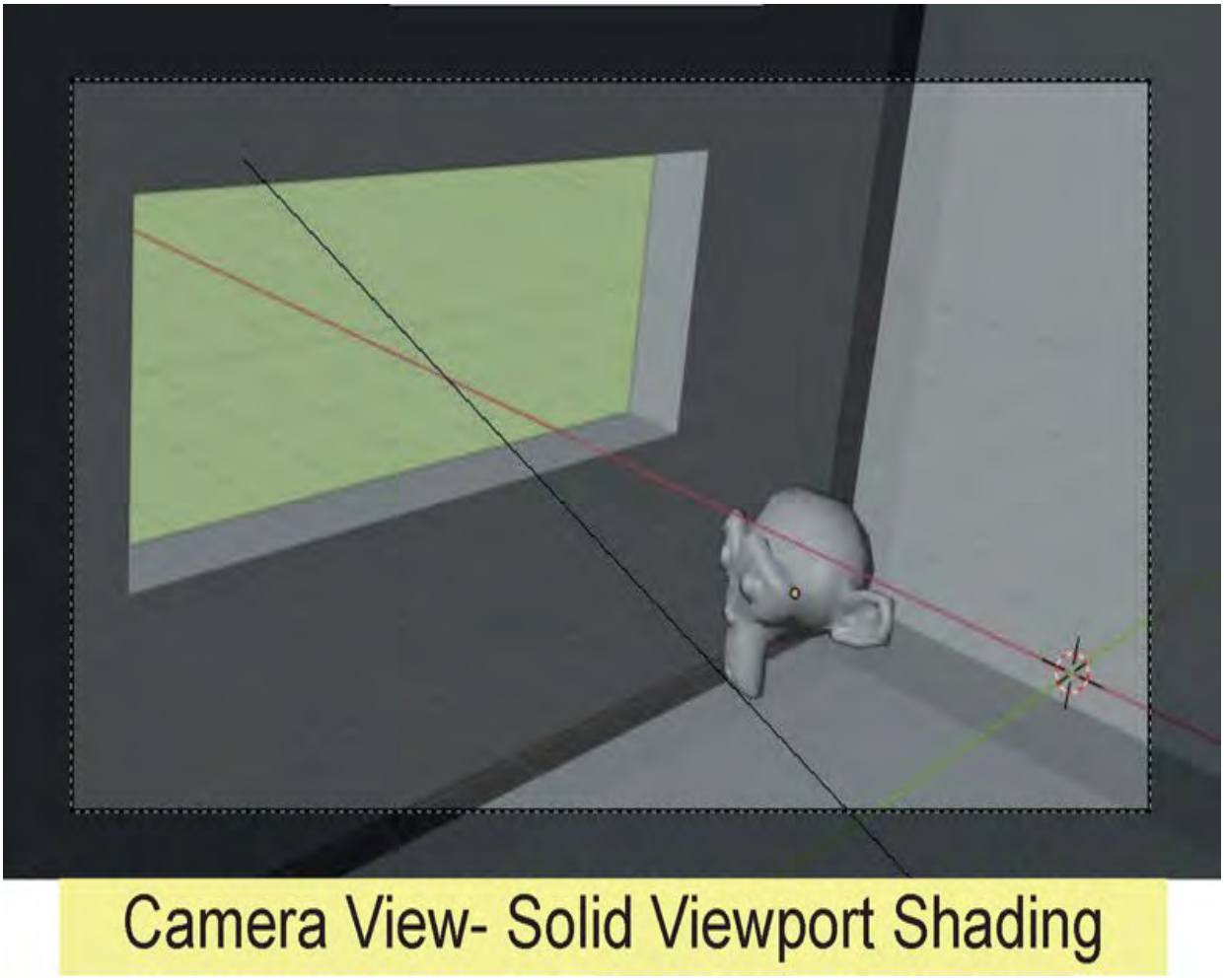


Figure 18.31

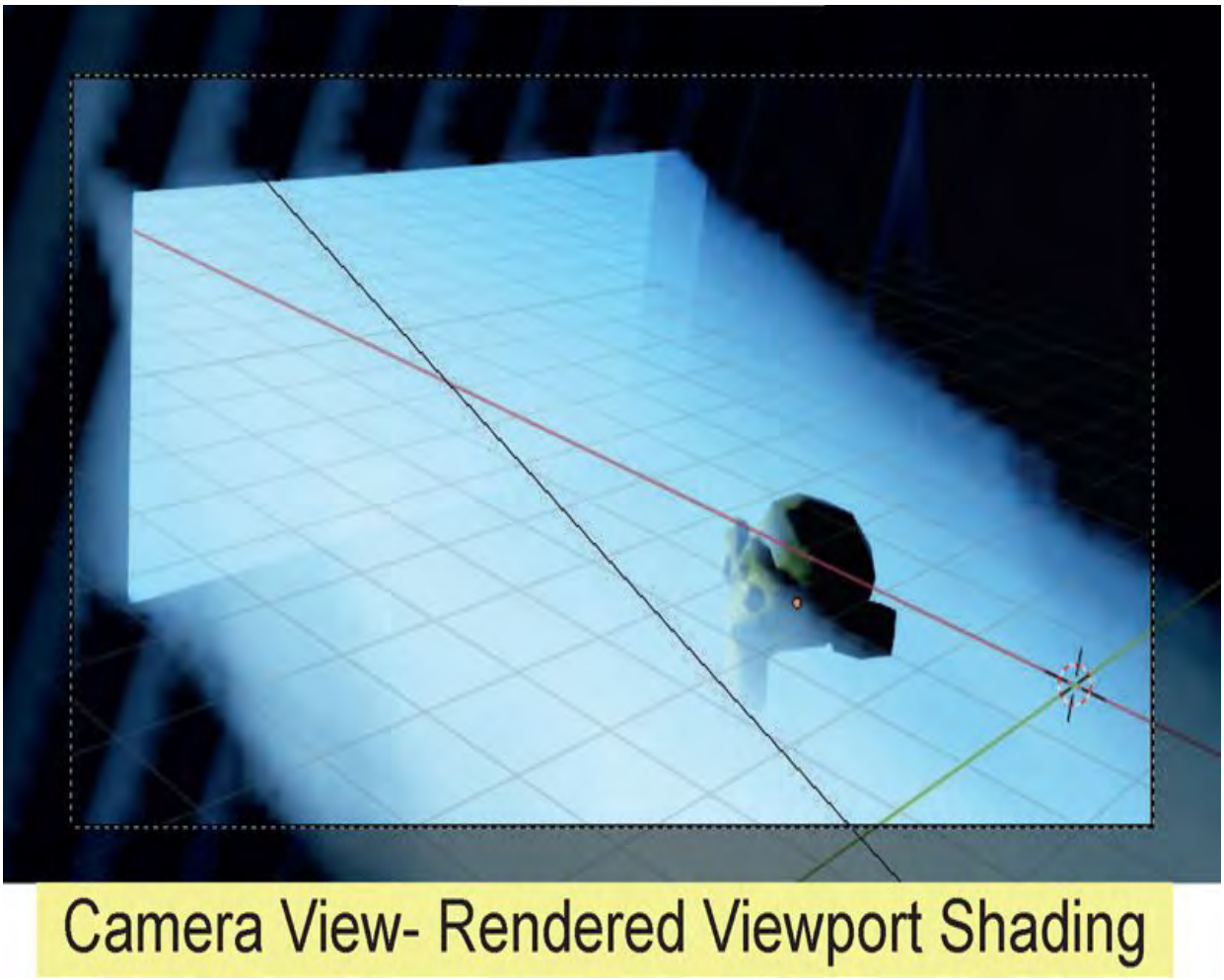


Figure 18.32

18.5 Compositing Nodes

Compositing Nodes (or composite, for short) allow you to create and enhance image files and video files. The contents of the Blender Scene can be the basis for the image or you may use an image already saved on your computer. A pre-saved image can be combined with other images or the Blender Scene to create a new image. Unlike Material and Texture Nodes, it is not necessary to have an Object selected in the 3D Viewport Editor or to have a Material applied to an Object. Of course, by default, any Object added to a Scene has the default material added to it even though this does not display in the **Properties Editor** until the **New** button is pressed.

To demonstrate the **Compositing Node Editor,** change the default Screen arrangement to that shown in Figure 18.33. To create this arrangement, start by changing the default **Layout Workspace** to the **Compositing Workspace.**

Compositing Workspace

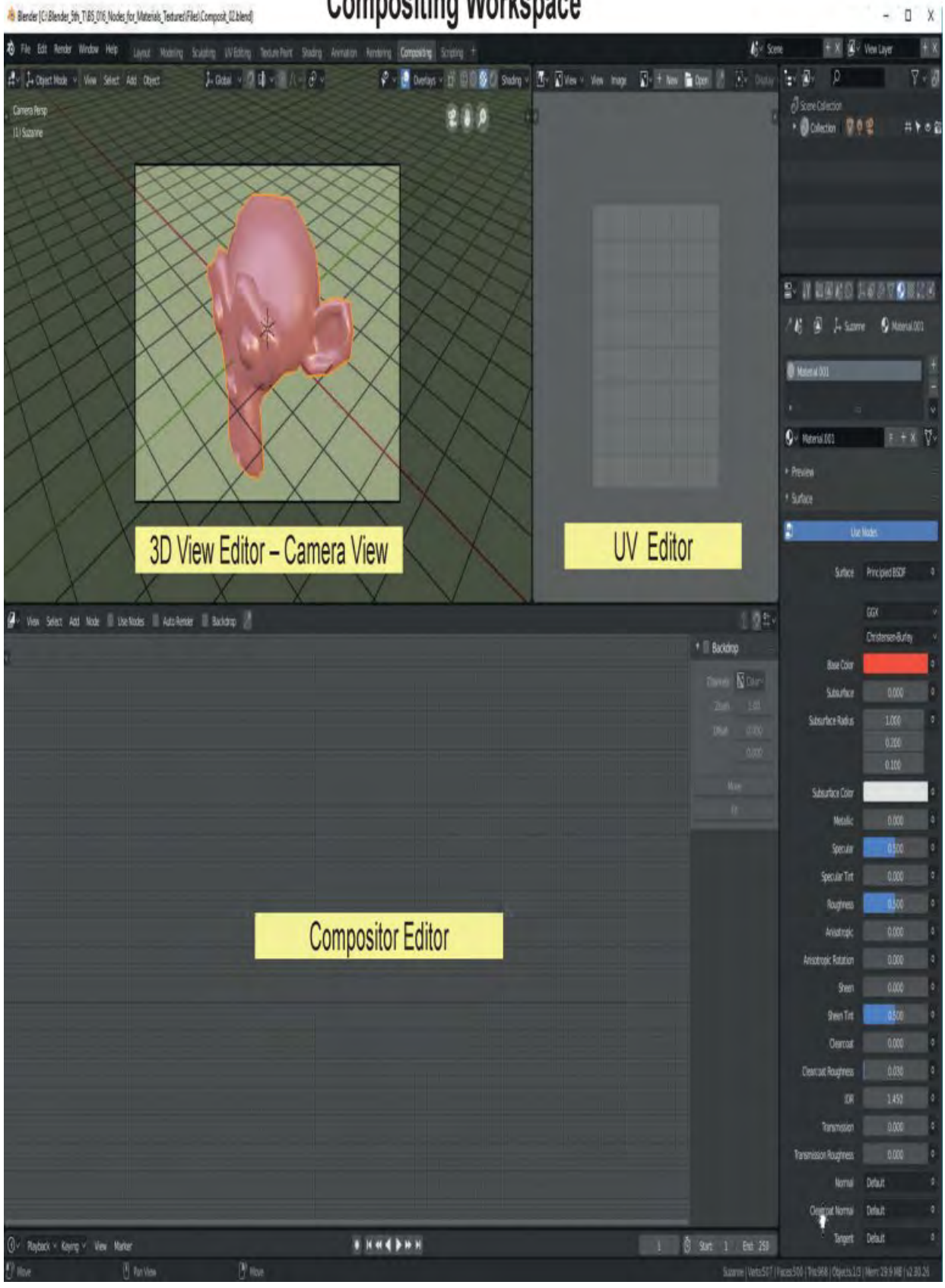


Figure 18.33

The Dope Sheet Editor at the bottom of the Screen will not be required in the demonstration since only still images will be used. Change the Dope Sheet to the **Compositor Editor,** change the existing Compositing Editor to the **3D Viewport Editor,** divide in two and make one half the **UV Editor.** Have the 3D Viewport in Rendered Viewport Shading Mode.

In this demonstration an **image rendered from the Scene** in the 3D Viewport Editor will be combined (composited) with an **image file downloaded** from the Internet.

In the 3D Viewport Editor, delete the default Cube and add a **Monkey Object.** Give it a nice bright Material (color). Change the **Viewport Shading to Material Preview Mode to see the Material.** Go into Camera View, Scale Monkey up to fill the view then deselect it in the 3D Viewport Editor (Figure 18.34).

Figure 18.34

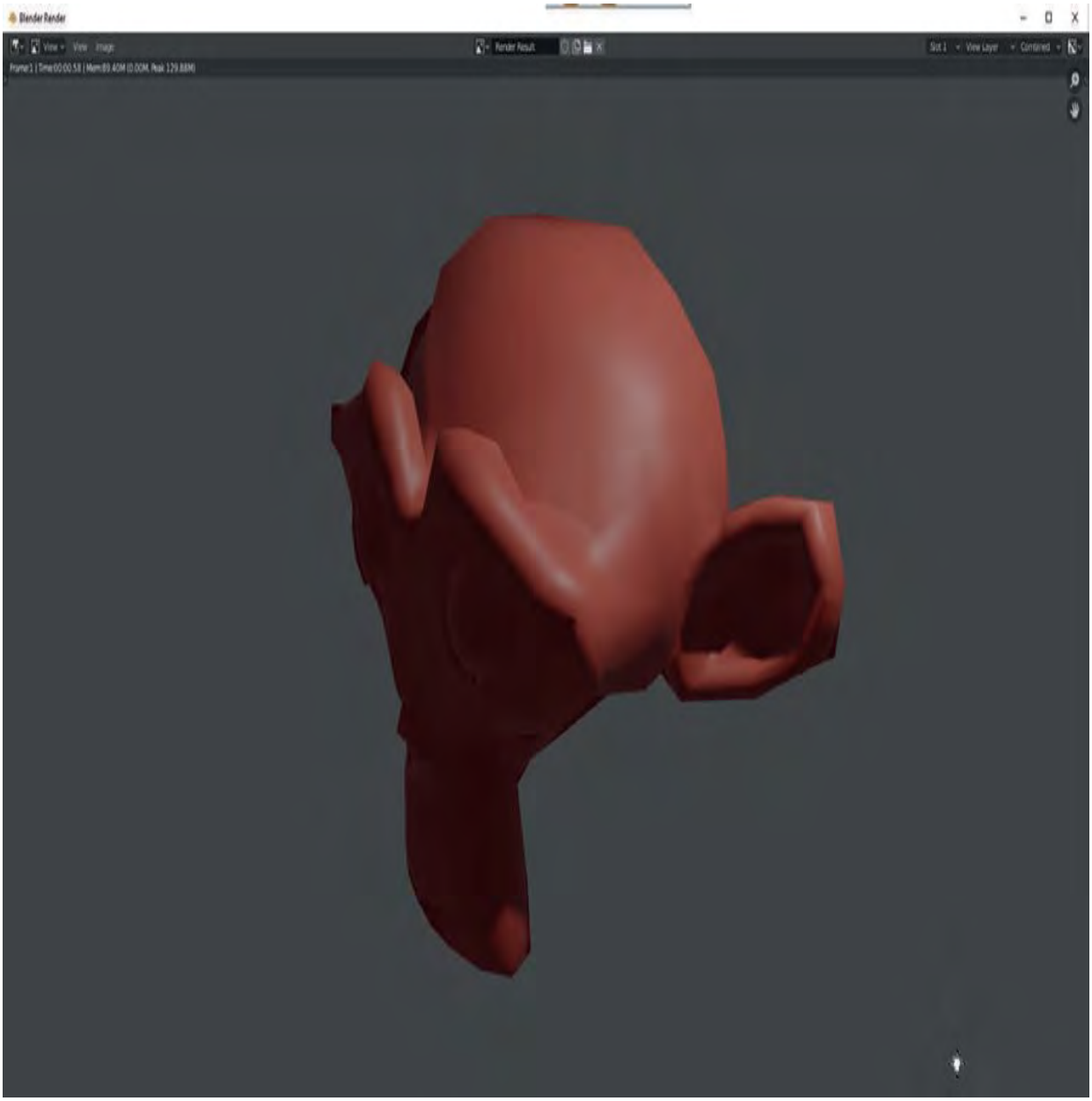


Figure 18.35 Rendered View Image Editor

Render a view of Camera View. In the Screen Header click **Render** and select **Render Image.** The image is Rendered and displayed in a full Screen version of the Image Editor (Figure 18.35). Cancel this Editor (click the X-upper RH of Screen).

In the **UV Editor** click on the **Browse Image to be linked** button in the Header and select **Render Result** to display the rendered image of the Monkey (scroll out) (Figure 18.36).

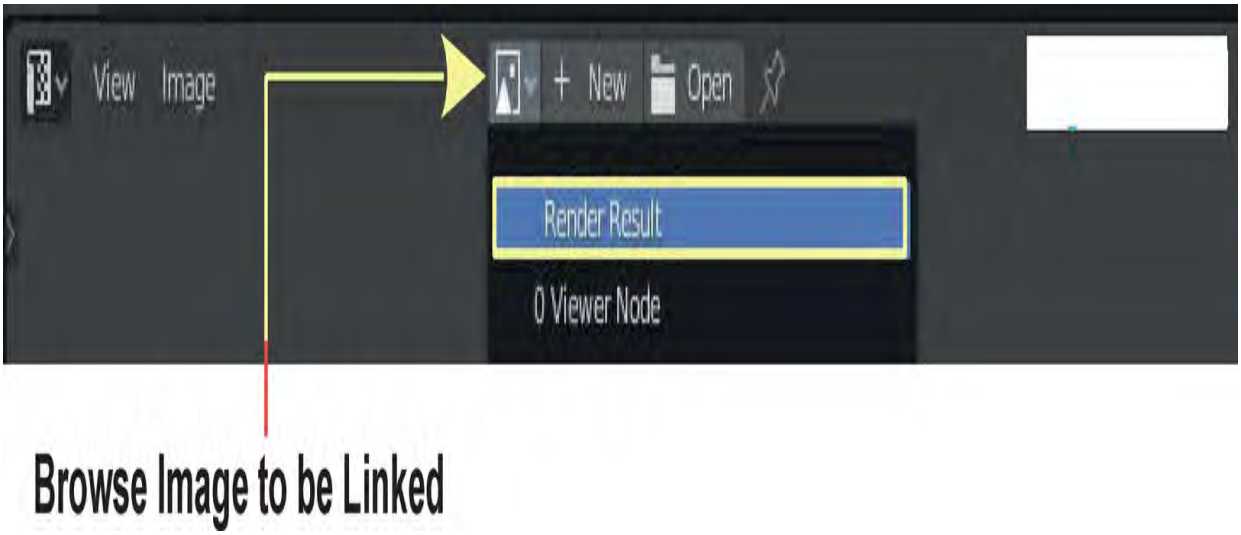
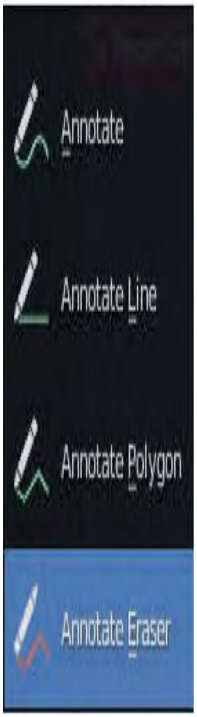


Figure 18.36

To combine the Rendered Image with another file click **Use Nodes** in the Compositor Editor Header (Figure 18.37).



With the Mouse Cursor in the Compositor Editor press the T Key to display the Tool Panel at the LH side of the Editor containing the Annotation Tool and the Link (Noodle) Cut Tool.

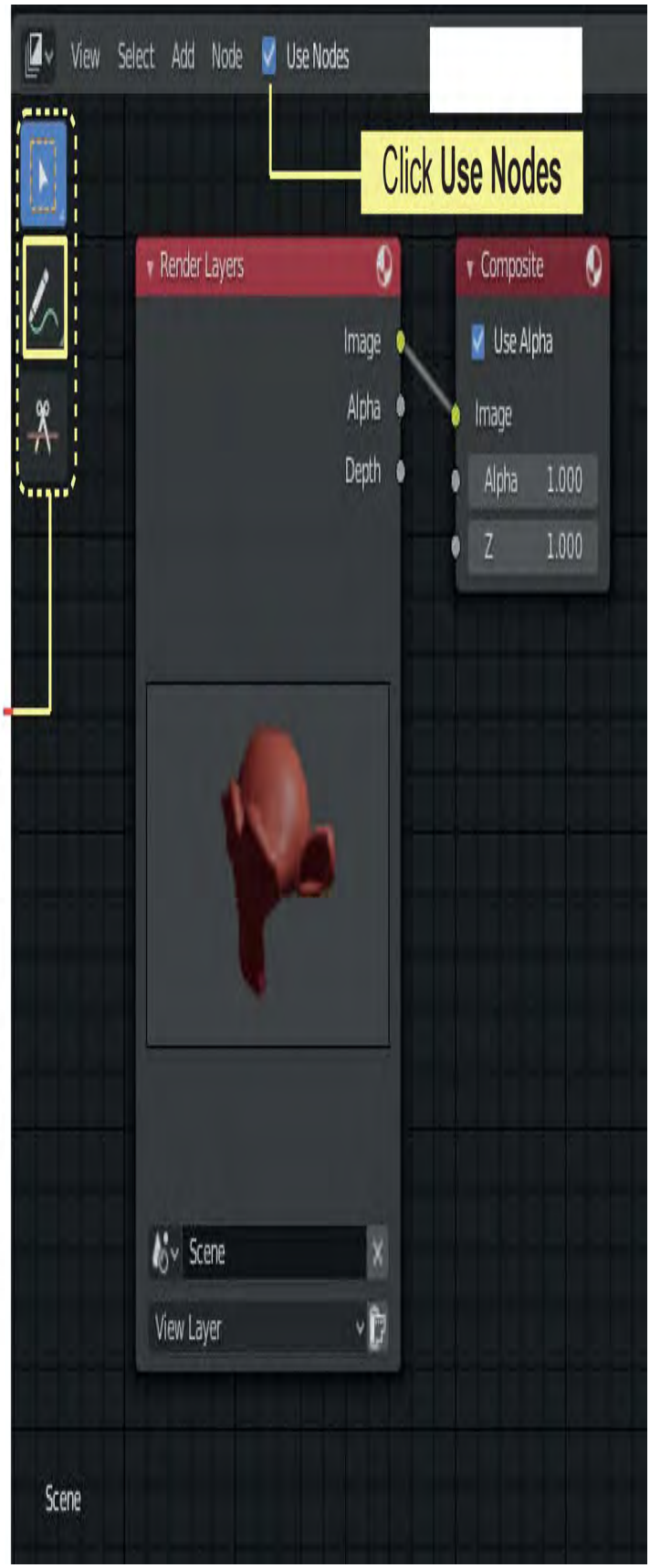


Figure 18.37

Mouse over on the Annotation Tool, click and hold for options.

Clicking Use Nodes displays the Render Layers Node connected to the Composite Node. A view of the Rendered Monkey shows in the Render Layer Node (Figure 18.37).

In the Header click Add and enter an **Input - Image Node** and a **Color - Mix Node.**

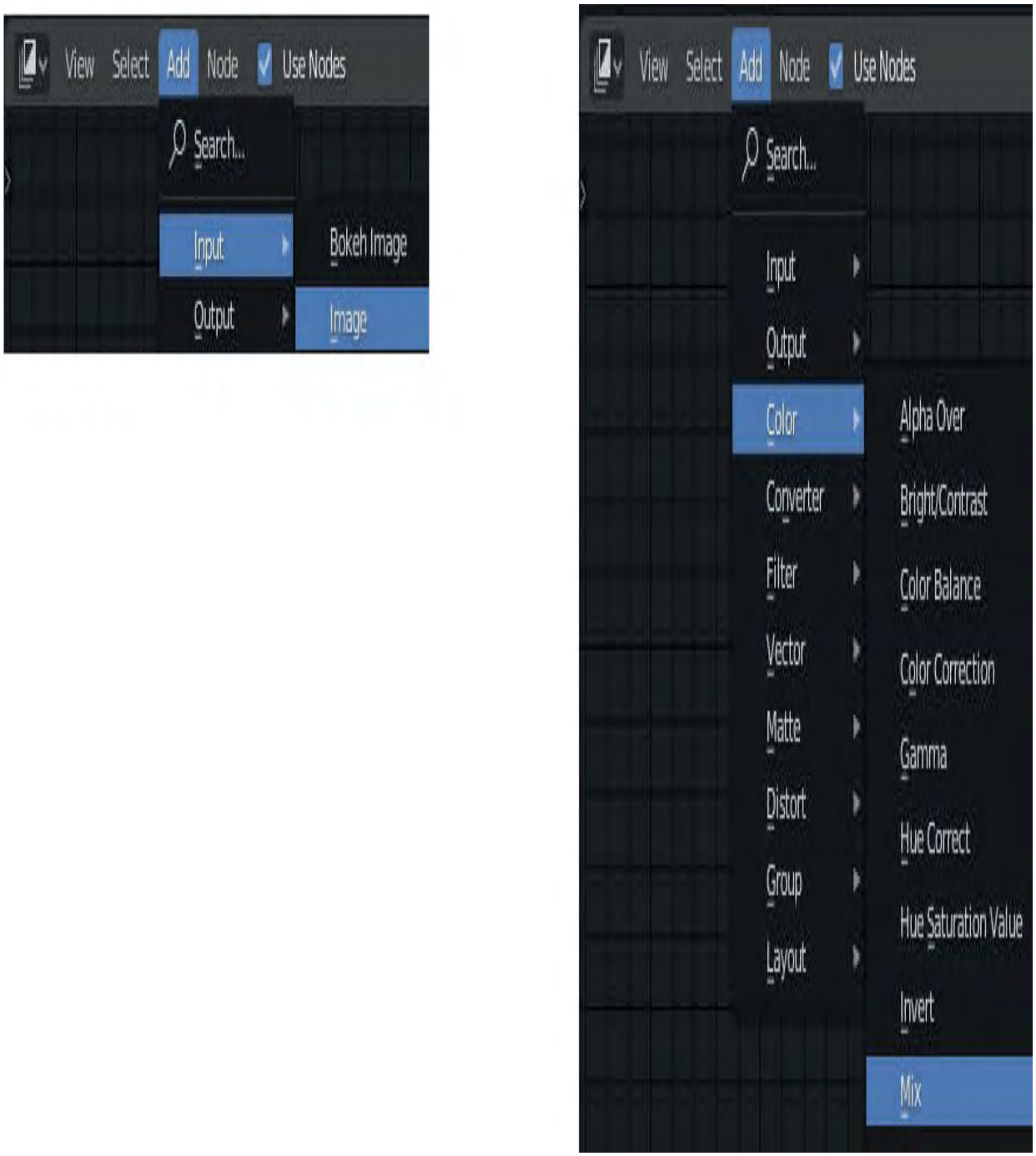


Figure 18.38

Arrange and connect the Nodes as shown in Figure 18.39.

In the **Image Node** click on Open, navigate in the File Browser that opens and select the downloaded image.

Render Layers
Image
Alpha
Depth
Experiment with the options.
Composite
Use Alpha
Image
Alpha: 1.000
Z: 1.000
Mix
Image
Mix
Clamp
Fac: 0.300
Image
Image
Adjust Factor
To show Image
evening-55067.png
Image
Alpha
Depth
Scene
View Layer
UV Image Editor
Click to open File View.
navigate to an Image.
eveni..
Single Image

Figure 18.39 Node Arangement

The selected image displays in the Image Node. In the **Mix Node** set the slider Fac: to approximately 0.300 and the Monkey Image in the Render layer is combined with the downloaded image and displayed in the UV Editor.

Render and save the new Image (Press F12 — Image Save).

Summary

The samples of Node Usage provided are, just that, a sample. The sample is intended to make you aware of how Nodes are used. There are dedicated publications on how to use Nodes and the results that can be achieved are limitless.

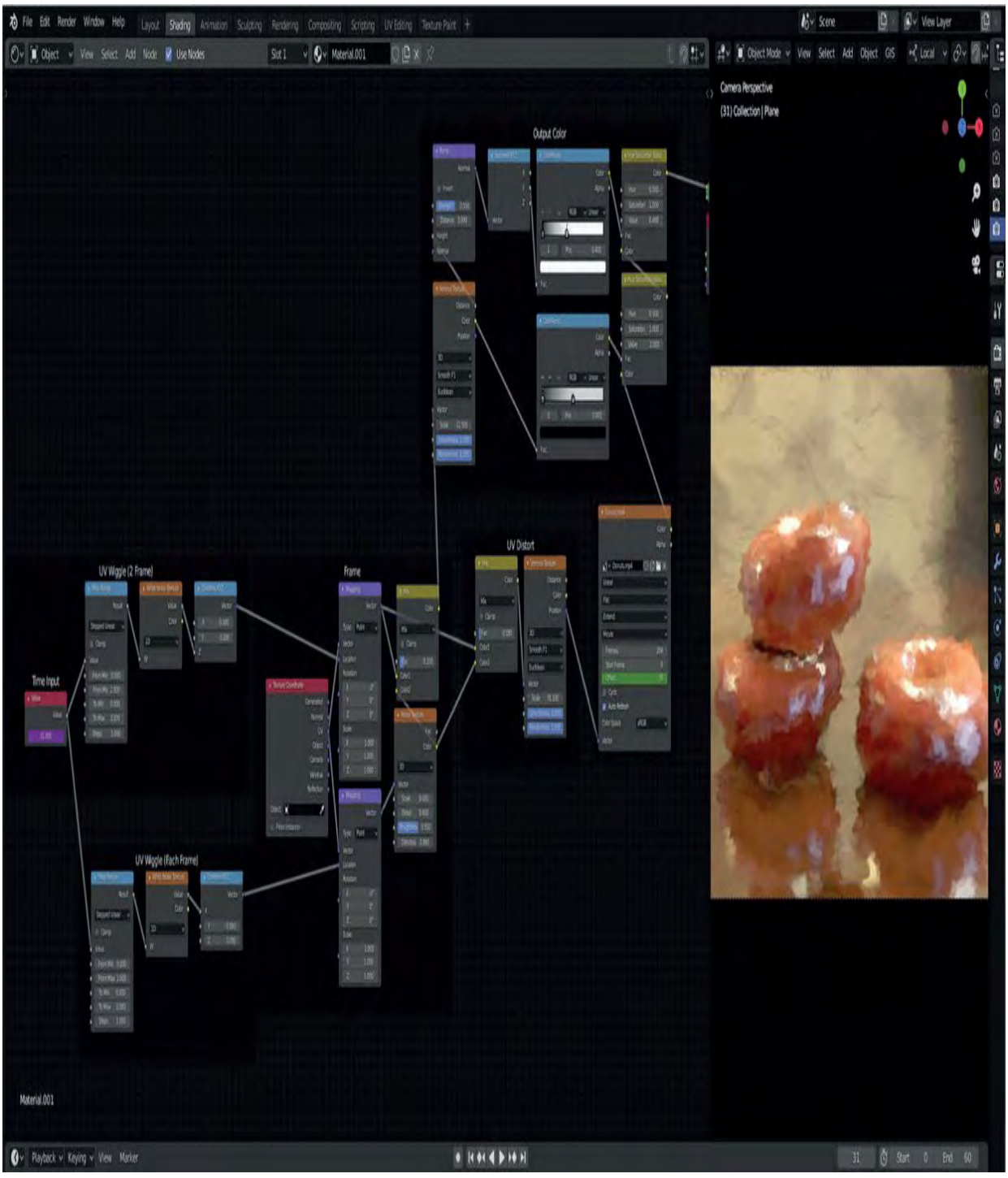

Output Color
UV Wiggle (2 Frame)
Frame
UV Distort
Time Input
UV Wiggle (Each Frame)
Material.001

Figure 18.40

19 Rendering

19.1 Rendering
19.2 Properties Editor Output Properties
19.3 The Dimensions Tab
19.4 The Output Tab
19.5 Rendering a JPEG Image
19.6 Rendering a Movie File
19.7 Video Playback
19.8 Video Codecs
19.9 Making a Movie

Rendering is the process of converting the Blender file information into an Image file or a Video Movie file. In practise this entails taking the data producing what you see in **Camera View** in Blender and converting it into **Image** or **Video** file format. The conversion may then be viewed as a digital still image or in the case of a video file, played on a variety of media devices depending on the format chosen. Controls and settings for Rendering are located in the **Properties Editor**

Note: There are three separate rendering systems in Blender. One is the Blender **Eevee** system, the second is the **Cycles** system and the third is **Workbench.**

This chapter explains the basic render procedure using the default **Eevee** system limiting the discussion to producing a still image file or video file.

Eevee is a fully-featured PBR (physically based-rendering) engine for real-time visualization. with advanced features such as volumetrics, screen-space reflections and refractions, subsurface scattering, soft and contact shadows, post-processing effects such as ambient occlusion, depth of field, camera motion blur and bloom, to name a few (definition from the Blender VViki).

As you work with **Eevee** engaged, what you see in **Camera View** with **Rendered Viewport Shading Mode** active, is what you get when the Blender file is converted to an image file.

19.1 Rendering

Rendering, in the practical sense, converts the data producing what is seen in **Camera View** in the 3D Viewport Editor into a still Image or in the case of an animation into a Video Clip.

To Render a **preview** of a still image press the **F12** button on the Keyboard. An image is Rendered and displayed in a full Screen version of the **Blender Render, Image Editor.** Press **Esc** to cancel. To Render an animation you would press **Ctrl + F12.** These Render options are also located in the Render button in the **Screen Header. Note:** Performing either option displays a **preview only.** To save the preview you are required to perform a secondary save function.

With the **Eevee Render Engine** active (Figure 19.1) it is not necessary to Render to see what you get in an image. With Eevee engaged, the 3D Viewport Editor in Camera View, **Rendered Viewport Shading Mode** you see the equivalent of a Render in the 3D Viewport Editor. You will see the Midplane Grid and the Scene X and Y Axis unless cancelled in **Overlays.**

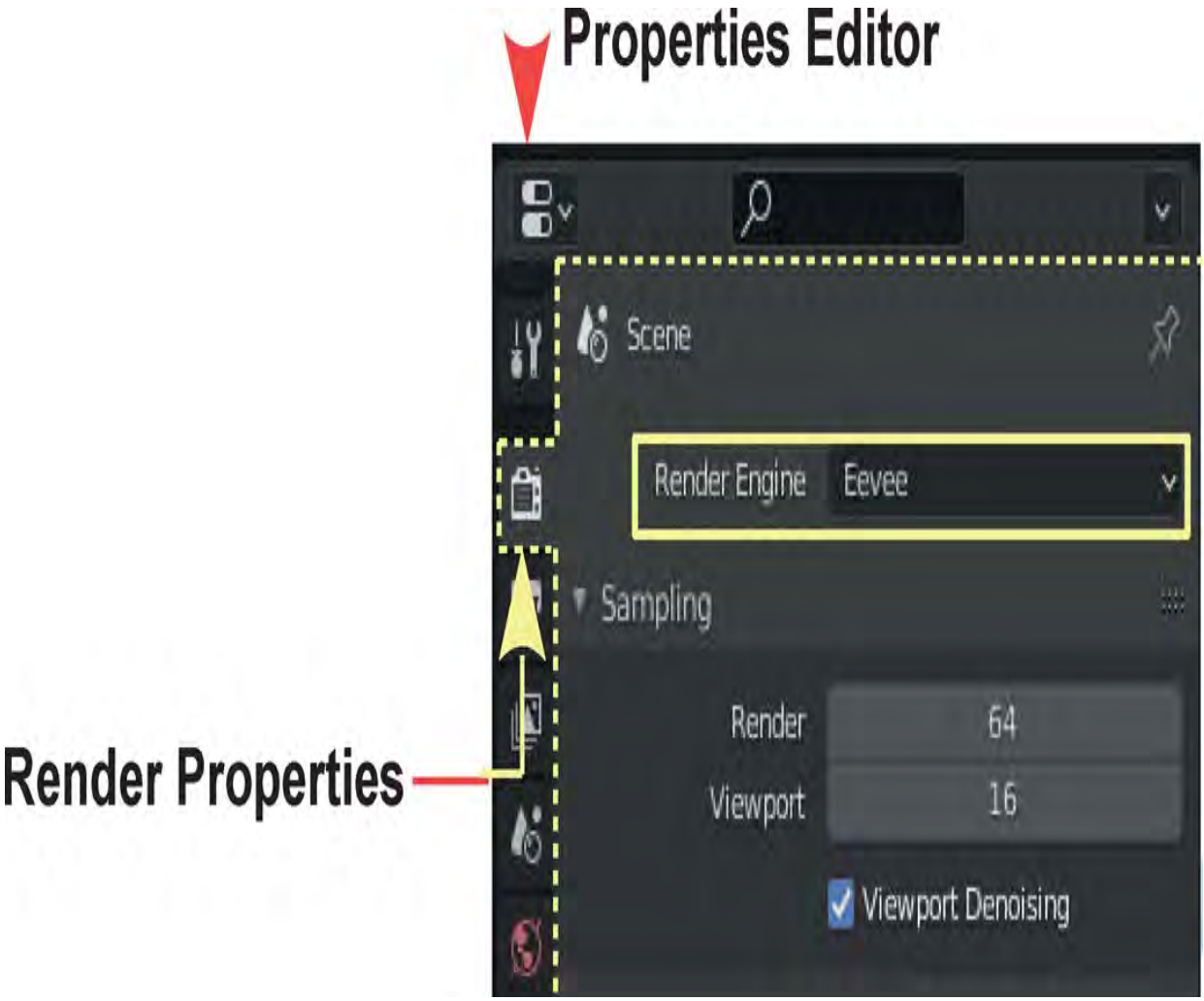


Figure 19.1

Note: Rendering a still image does not save a file to the computer.

With the default settings in the **Properties Editor, Render Properties,** Rendering a still image merely displays a preview of the Render in the **Blender Render, Image Editor.**

Rendering an animation, however, saves an image for each Frame in the animation and by default, saves to the **/tmp** directory as seen in the **Properties Editor, Output Properties.** On a Windows 10 Computer this means the files are saved in Local Disk (C:)\tmp.

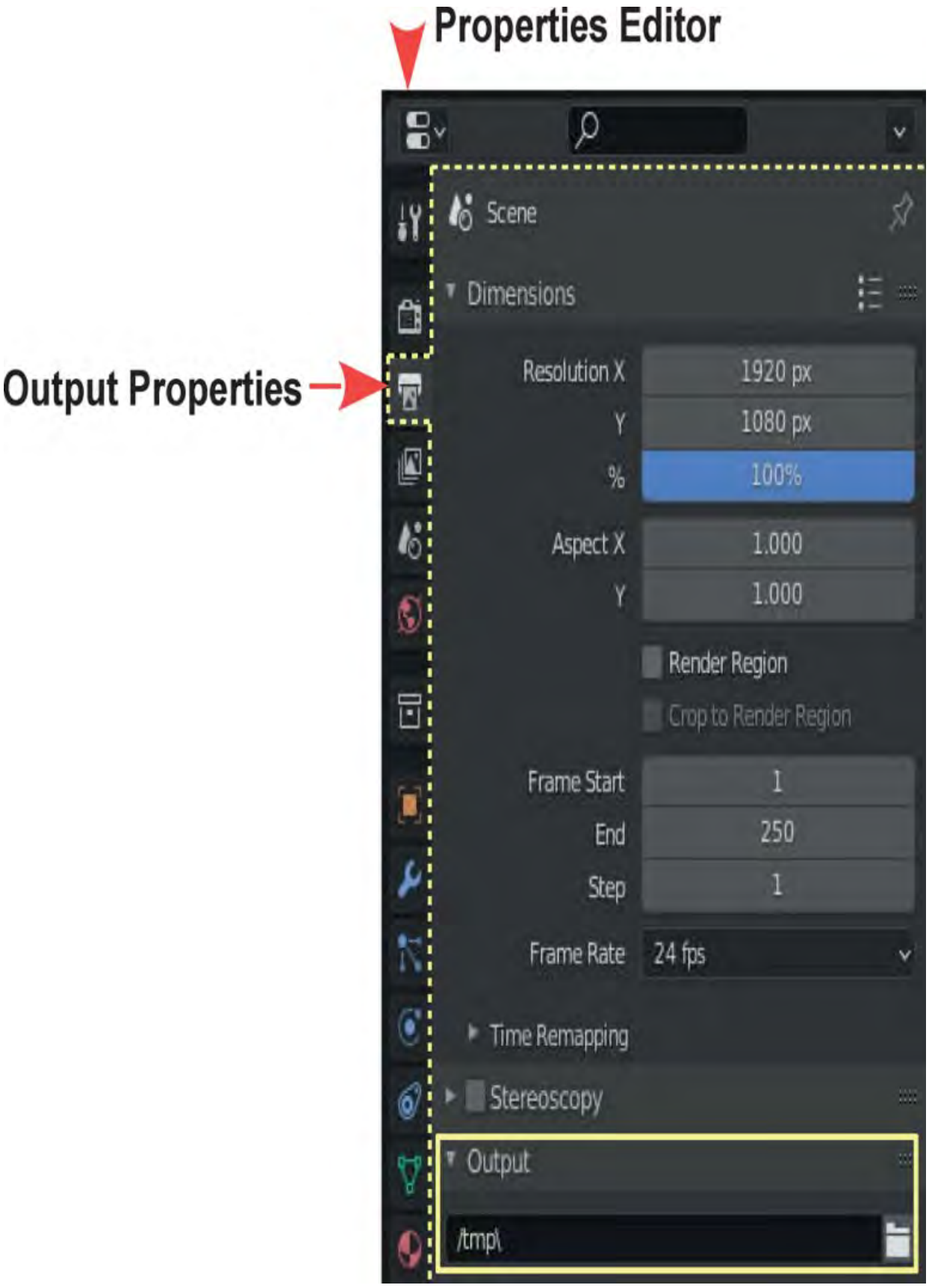

Properties Editor
Output Properties
Scene
Dimensions
Resolution X
1920 px
Y
1080 px
%
100%
Aspect X
1.000
Y
1.000
Render Region
Crop to Render Region
Frame Start
1
End
250
Step
1
Frame Rate
24 fps
Time Remapping
Stereoscopy
Output
/tmp\

Figure 19.2

The detailed output from the Render process is controlled in the **Properties Editor, Output Properties.**

The Render controls are shown here to make you aware of their existence. The controls are arranged in Tabs. Opening a Tab displays the controls for that particular subject which in some cases requires activation by checking the button preceding the subject name.

When Rendering, the computer is converting the data producing the Screen display into a format for an Image display. In doing this, it processes in a series of passes, which in each pass improves on the quality of the previous pass. How many passes performed is referred to as the number of **Samples.** As you see in Figure 17.1 the default number of Samples is; **Viewport 16** (what you see in the Viewport) and **Render 64** (what you get when you press F12).

19.2 Outpub Properties

In accepting the default number of Render Samples and Render Settings, to produce an image or video, all you have to do in practice is specify the **Resolution** and where you want the file saved. This is done in the **Output Properties** (Figure 19.3)

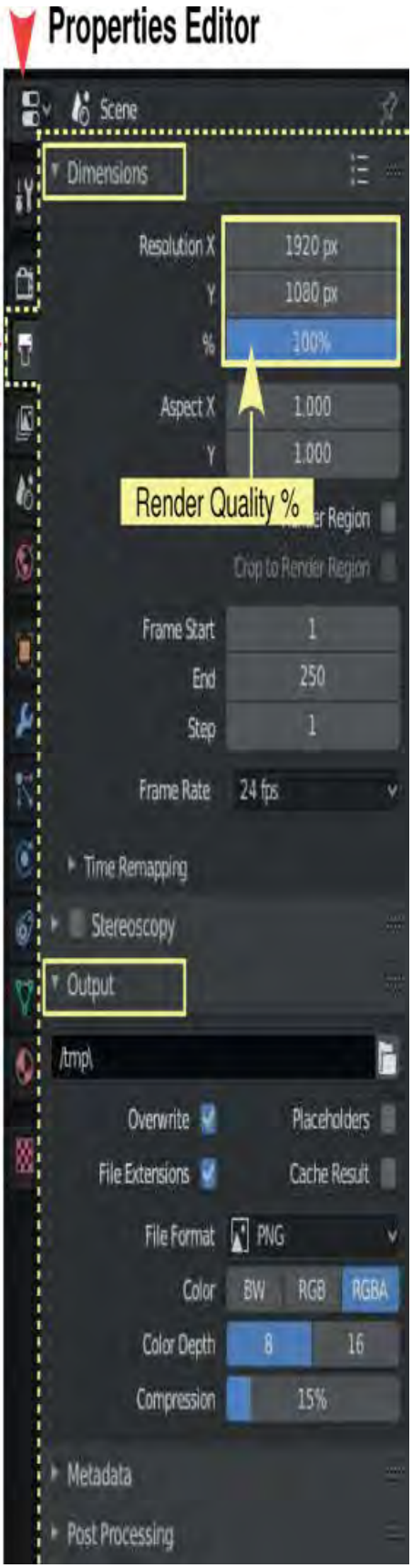


Figure 19.3

In the Output Properties the **Dimensions Tab,** and **Output Tab** are expanded.

You may consider the expanded tabs as the basic controls for setting the rendering process.

19.3 The Dimensions Tab

The **Dimensions Tab** (Figure 19.3) is where you tell Blender how big to make your image, the shape of the image, the quality of the image (Resolution) the shape of the pixels (Aspect Ratio) and in the case of an animation where to start and stop rendering and how fast you want it to play back when finished (Frame Rate).

Resolution

X : The number of pixels wide in the display. **Y:** The number of pixels high in the display.

Pixels are the tiny little rectangles which make up the display on the computer or television screen.

> **Note:** The default **Resolution** 1920 x 1080 equates to the HDTV 1080p **Preset.**

The percentage slider sets the quality of the render. The default is 100% which scales the resolution. If the value were 50% although the resolution is set at 1920 x 1080 the render would be a preview at 960 x 540. Since rendering takes time this is a way of seeing your image or movie as a preview prior to a final render and therefore saving time. For the final render you set the slider to 100%.

Aspect (Aspect katio)

The aspect ratio refers to the shape of the pixels. The default ratio X:1.000, Y:1.000 (1:1) is for computer monitors which have square pixels. TV screens have rectangular pixels so you have to set a ratio for the appropriate format i.e. HDV NTSC 1080p for America the ratio is 4:3 and HDV PAL 1080p for Europe is also 4:3 but TV PAL 16.9 the ratio is 16:11.

Aspect ratios are very confusing. Figure 19.4 is offered as a guide.

Figure 19.4

Frame Size	**Aspect Ratio**	**Description (note these are only the most common formats)**
1920 x1080	16x9	1080p/i
1440 x1080	16x9	1080i **(Most HDV use this format)**
1280 x720	16x9	720p
852x 480	16x9	480p
720x 480	**4:3**	DV NTSC (when the pixels are square it is actually 3:2)
720x 480	16:9*	DV NTSC / Anamorphic* *I* Wide Screen (non square pixies)
720) (576	5:4	DV PAL
640x 480	**4:3**	a ration suitable for square size pixie multimeida video.
640x 360	16:9	a ration suitable for square size pixie multimeida thats widescreen.
480x 360	**4:3**	Multimedia large (480x360 75%(640x480))
480x 270	16:9	Multimedia Large (similar to Apple's large move trailer standard 480x272) (480x270 75%(640x360))
320x 240	**4:3**	Multimedia Large
320x 180	16:9	Multimedia Large I Wide Screen
240x 180	4:3	Multimedia Small

Frame Size	Aspect Ratio	Description (note these are only the most common formats)
1601(120	4:3	Thumbnail
1600x1200	**4:3**	Computer Display
1280x1424	4:3	Computer Display
1152x870	4:3	Computer Display
1024x768	**4:3**	Computer Display
800x600	**4:3**	Computer Display

Aspect Ritlo

The ratio between the length and width of video images. **NTSC, PAL,** and **Secam** formats use a **4:3** aspect ratio. Newer, more advanced formations such as **HDTV** (High Definition Television) use a much wider aspect ratio of **16:9.**

- Television is 4:3
- Widescreen TV 16:9
- 35mm Film 1.65:1
- 70mm Film 2.0:1

Frame Start – End – Step (Frame Range)

This is fairly self explanatory and shows the start frame (Frame Start) and end frame of the animation and the **Steps** which means which frames to render. **Frame Steps: 1** means render every frame, **Frame Steps: 2** would mean render every other frame **Frame Steps: 3** would mean render every third frame etc.

> **Note:** Pressing **F12** with an animation paused renders an image of the single frame in the animation where it is paused.

Frame Rate The playback speed of the animation expressed in Frames per second. The selection menu provides options for a variety of formats. 24 Frames per second is the default setting (25 FPS for PAL TV European format and 30 FPS for NTSC TV US format – These frame rates are approximate and vary with the actual **Render Preset** selected).

19.4 The Output Tall

In the **Output tab** (Figure 19.2 – 19.5) you set options to tell Blender where to save your render and the file format required.

Output
/tmp\
Overwrite
Placeholders
File Extensions
Cache Result
File Format
PNG
Color
BW
RGB
RGBA
Color Depth
8
16
Compression
15%
File Format
Image
BMP
Iris
PNG
JPEG
JPEG 2000
Targa
Targa Raw
Cineon
DPX
OpenEXR MultiLayer
OpenEXR
Radiance HDR
TIFF
Movie
AVI JPEG
AVI Raw
FFmpeg video

Figure 19.5
Figure 19.6

By default Blender will save your render to the **temporary** folder on your hard drive as seen by the **/tmp** notation in the output file address bar. On this computer this is C:\tmp\. You can choose a different location by clicking on the folder icon at the end of the bar and navigating in the file browser that opens.

Blender will save your render in a variety of fil formats. The default format is **PNG** (PortablE Network Graphics). Where you see this in the Output Tab is a selection menu for choosinc alternative formats (Figure 19.6). In the menu yo will see that the options are in two categories **Image** and **Movie.** Image types such as PNG [illegible of text in source] JPEG produce a render of a still image in tha[illegible of text in source] particular file type. Selecting one of the Movie[illegible of text in source] options produces a render of an animation in [illegible of text in source] compressed movie file such as AVI Raw or FFmpeg.

Note: With an Image file selected a render of an animation will consist of a series of images of each frame of the animation. Although this takes up a lot of room in a folder it is an acceptable method of producing a video file.

Note: The default **Image** type PNG with the RGB color scheme and **Compression** ratio of 15%. Some formats can compress images to use less disk space, for instance; Lossless PNG or JPEG.

19.5 Rendering a JPEG Image

1. When you have created a scene in the 3D Viewport Editor and decide that you wish to save an image of what you see in the **Camera View** go

to the **Properties Editor, Output Properties.** For the time being leave all the default settings just as they are. In the **Output Properties, Output Tab,** click on the selection menu where you see **PNG** and change to **JPEG.**

2. With your mouse cursor back in the 3D Viewport Editor, press F12 on the keyboard to render the **preview** of camera view. The preview displays in the Image Editor (Blender Render). In the Header click **Image** and select **Save.** Enter a name for your image in **Blender File View** and click **Save As Image** at the bottom of the panel.
3. If you have opened a previously saved image and made changes you will press Save As (Shift + Alt + S Key). The File View displays (Figure 19.7) which allows you to rename your image file and navigate to a folder on your hard drive where you wish to save the modified JPEG image.

In Figure 19.7 the image file has been named **Camera_View_Rendre.jpg** and is saved to **A_My_Blender_Stuff** folder on the hard drive. With the information entered as shown, click **Save As Image** in the lower RH corner of the File View window. **Note:** If you have an animation sequence paused you can render and save an image of the animation frame.

Blender File View
Volumes
WINDOWS (C:)
RECOVERY (D:)
DVD RW Drive (E:)
System
Documents
Desktop
C:\
Name
Date Modified
Size
$RECYCLE.BIN
31 Oct 2017 15:11
$SysReset
02 Feb 2018 14:47
A_Blender_6th_2.81_Otd
A_BTD_Components
A_BTD_Helens_Stuff
31 Oct 2019 10:33
A_My_Blender_Stuff
05 Dec 2019 06:09
A_Nikon_Photos
14 Nov 2019 11:05
Folder to Save File
File Format
JPEG
Color
BW
RGB
Quality
90%
Save As Render
Copy
Filter Alembic files
Camera_View_Render.jpg
Save As Image
Cancel
Enter your file name
Click to Save Your Image
Save As Image
File Format
PNG
Color
BW
RGB
RGBA
Color Depth
8
16
Compression
15%
Save As Render
Copy
Note: If you have forgotten to change the file format from PNG to JPEG when you saved you can change it in the Panel at the upper LRH side of the File View window.

Figure 19.7

19.6 Rendering a Movie File

The title, **Rendering a Movie File** should possibly be named, **Rendering a Video Clip.** Movies are made by combining a series of video clips (short sections of video). Video clips are made from animation frames compiled into a sequence. Before you can render a video file you must have an animation sequence.

To demonstrate the process an animation will be created and rendered to video. Animation is covered in the next chapter so assuming you are working your way through the book chapter by chapter you will not have covered the topic at this stage. To create an animation work through the following instructions.

In the **3D Viewport Editor** go into **Camera View** (Num Pad 0) and Translate the default Cube back along the X Axis outside the Camera View (Figure 19.8 **Cube at Frame 1).** Press the **I Key** and in the drop down menu, select **Location.** In the **Timeline Editor** at the bottom of the screen click RMB on the vertical blue line, hold and drag the line to frame 10. In the 3D View Editor, Translate the Cube along the X Axis to outside Camera View on the opposite side. Press **I Key** again and select **Location.**

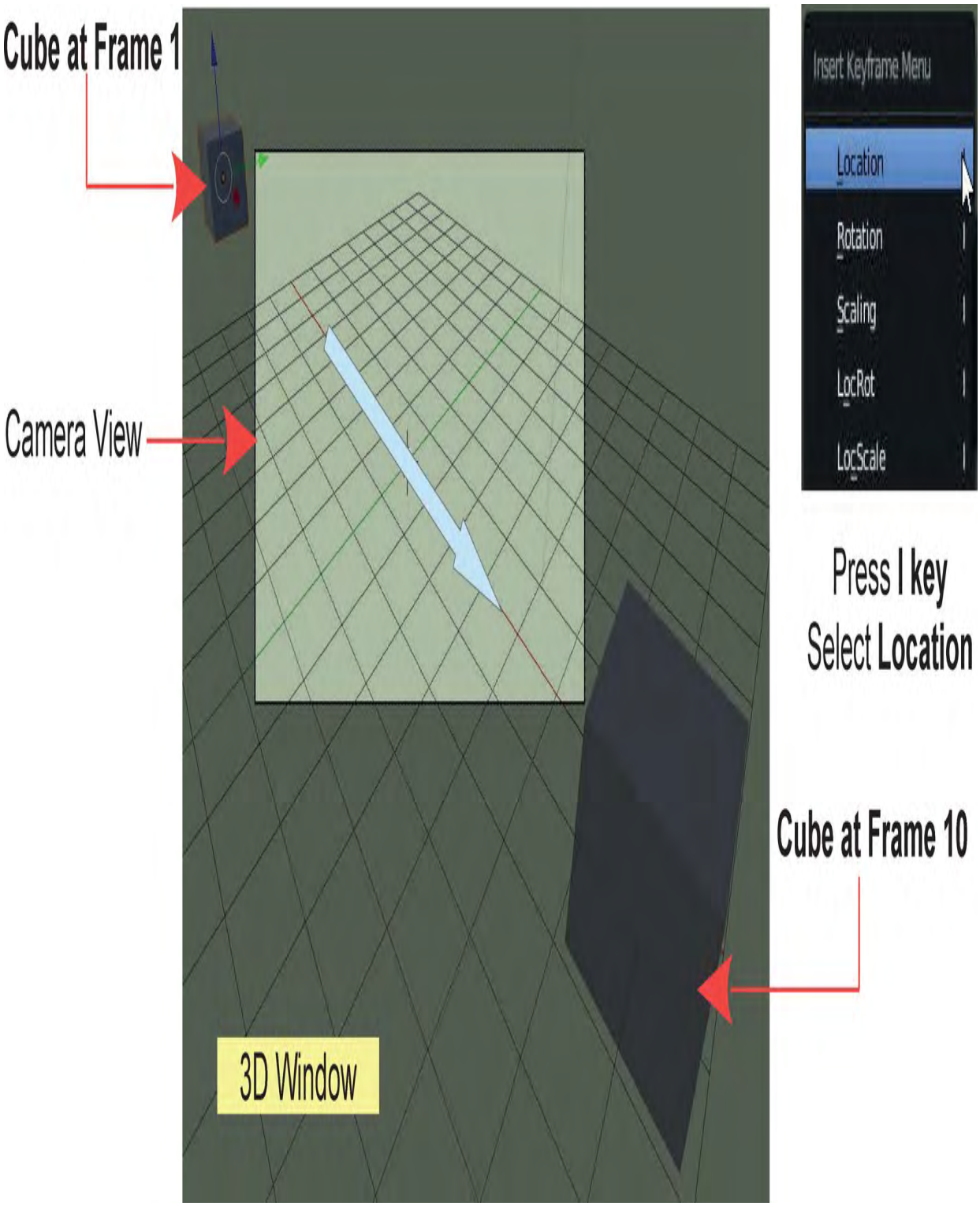

Cube at Frame 1
Camera View
Insert Keyframe Menu
Location
Rotation
Scaling
LocRot
LocScale
Press I key
Select Location
Cube at Frame 10
3D Window

Figure 19.8

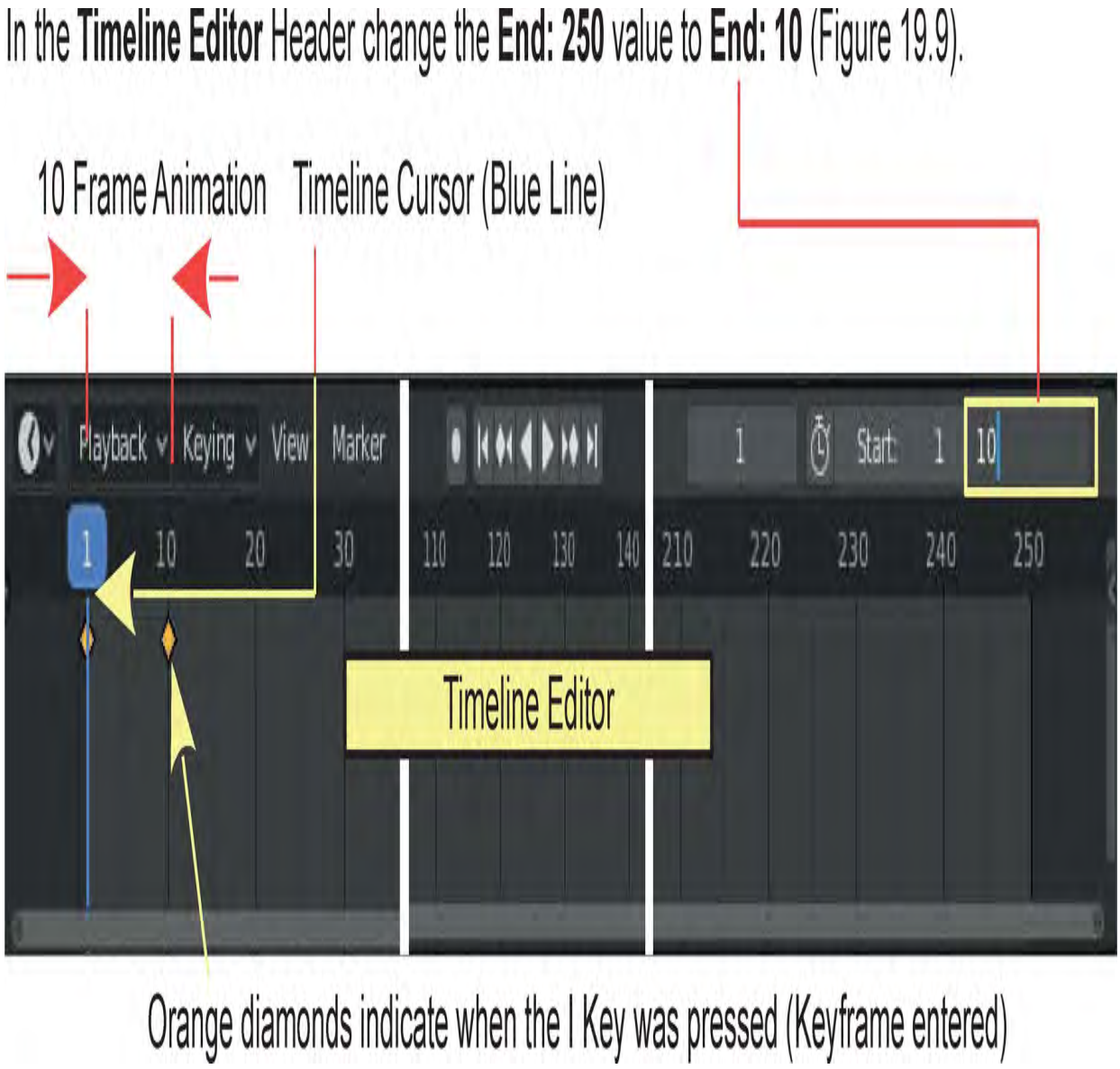


Figure 19.9

You have made an animation consisting of 10 frames. In the Timeline Editor click on the reverse arrows to set the animation at frame 1 (Figure 19.10). Click the start button to play the animation in the 3D Viewport Editor. The Cube zips across the Camera View and repeats the 10 frame animation over and over. Press **Esc** to quit.

Figure 19.10

OK! Now for the render.

In this demonstration use the default settings in the **Properties Editor, Output Properties.** The **Output Tab** is set to render a **PNG** file format and save it to the **/tmp** folder on the hard drive (Figure 19.11). If you haven't been messing with your hard drive and re-partitioning, the /tmp\ folder should be in **C:\tmp.**

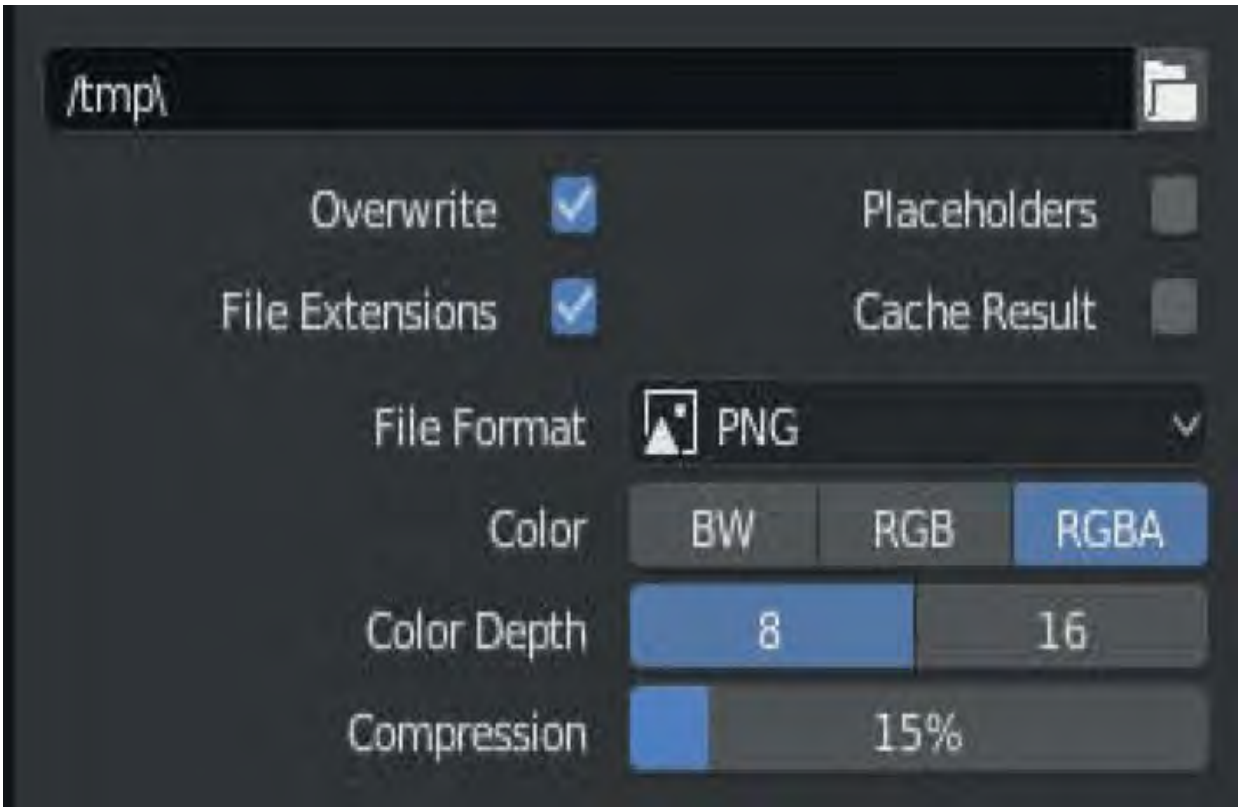


Figure 19.11

tmp stands for temporary. Programs use this location for storing information while they work therefore there could be files in the folder.

Before you do any rendering to the **/tmp** folder give it a clean out so you have a clean slate for this demo.

To Render the animation click **Render** in the **Screen Header** and select **Render Animation (Ctrl _ F12)** (Figure 19.12).

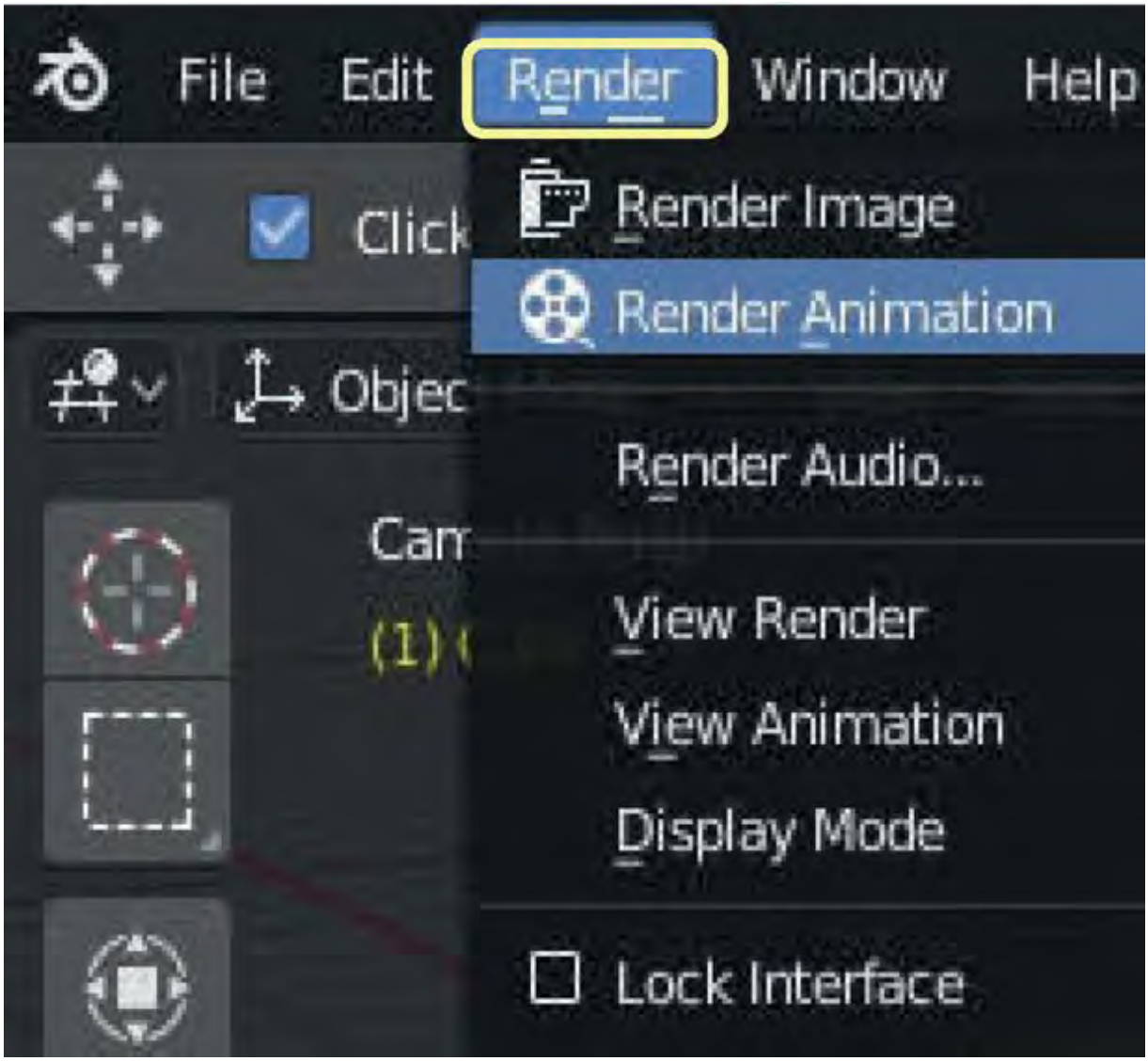


Figure 19.12

Blender will start making an image for each frame of the animation and save it to the **/tmp** folder. The reason for Rendering an image for each frame is that the default output file type is **PNG** which is an image file. The same thing would happen if you had JPEG selected. The Render is performed in the Image Editor where progress displays at the top of the Screen. When the render is finished Close the Image Editor. If you look in the /tmp\ folder when the render is finished you will see 10 image files (Figure 19.13).

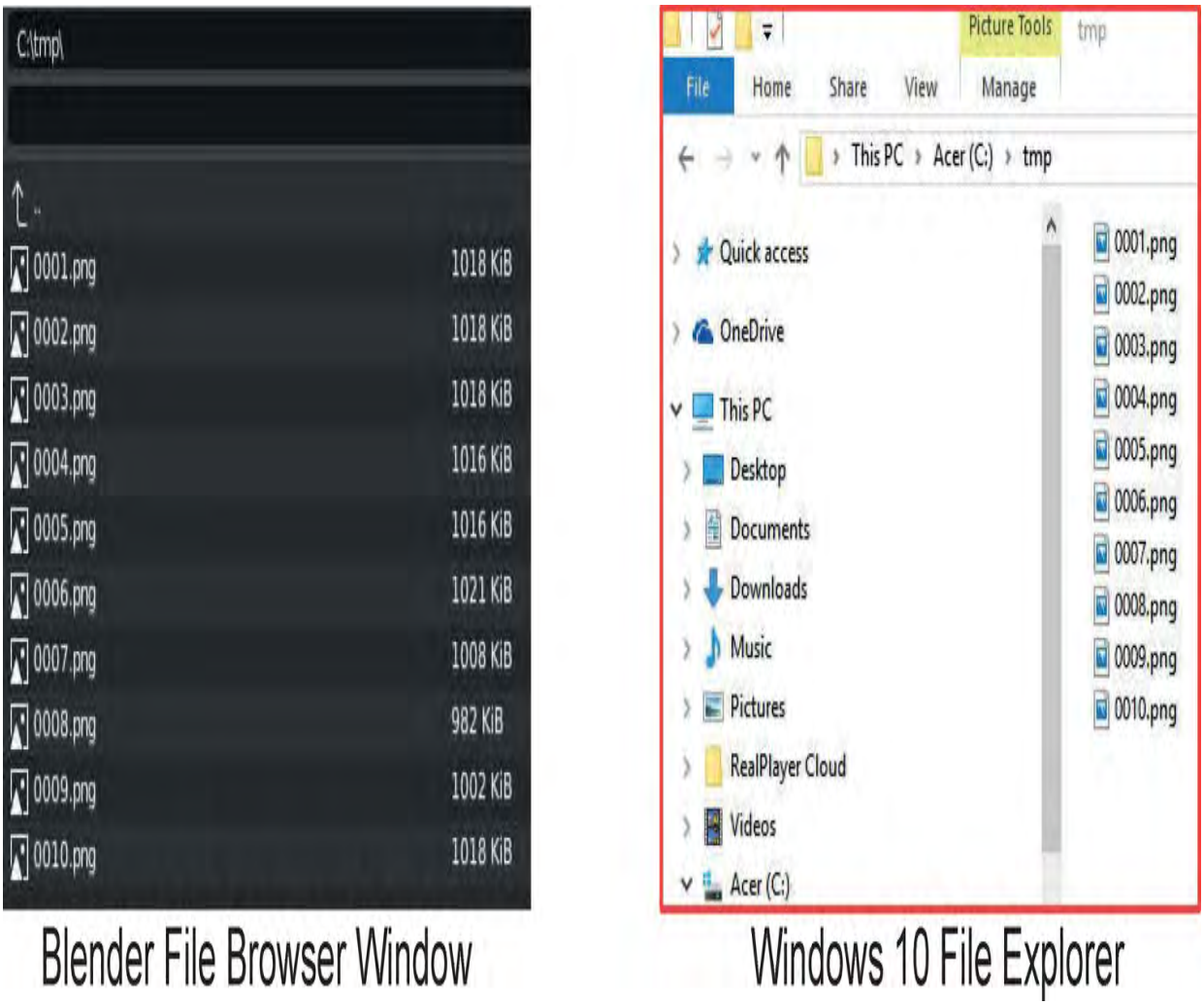


Figure 19.13

19.7 Video Playback

To playback your rendered animation from within Blender you go to the **Screen Header** and click on **Render** and select **View Animation.**

Playing the animation at this stage is simply cycling through the sequence of image files that has been created. Ten simple image files constitute a very very basic animation. Animations can run to thousands of image files which would accumulate and create a massive storage problem on your hard drive. To save space you render the animation sequence to a **Movie File.**

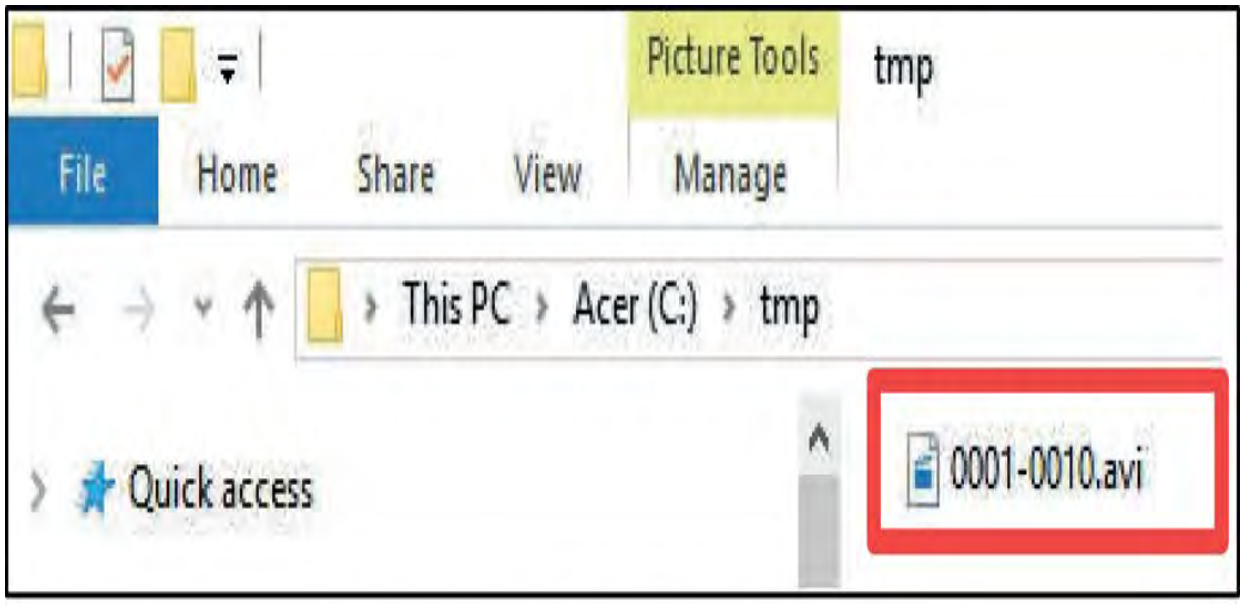
As an example take the same 10 Frame animation previously created and in the **Properties Editor, Output Properties, Output tab,** change the default **PNG** to one of the other **Movie** options, for example **AVI Raw,** which is a video file format (Figure 19.14). Video file formats compress the data from the rendered animation into a single file instead of the series of image files.

Figure 19.14

Before you Render with the video file format clean out the **tmp** folder again.

OK! With a clean slate go ahead and select **Render Animation** to start Rendering. When Blender is finished Rendering press **Esc** to exit the Image Editor.

Go look in your **tmp** folder and you will find a single video file (Figure 19.15). Press the **Play** button to replay the rendered animation in the inbuilt player. Since this is a video file you could also play it in some external application such as **VLC Media Player.** If you try this with the 10 frame animation pay attention since 10 frames plays pretty quickly. An external player only plays the file once and at 24 frames per second this is less than half a second of video.



Windows 10 File Explorer

Figure 19.15

19.8 Video Codecs

In the preceding example, changing the default **PNG** file format to **AVI Raw,** elected to use the **AVI Raw Video Codec** which tells the computer how you

want your animation data encoded. There are many many video codecs to chose from and simply selecting a codec type in Blender doesn't necessarily mean that you will get the result that you want. You must have the **Codec** installed on your computer.

A Codec is a little routine that compresses the video so that it will fit on a DVD, or be able to be streamed over the Internet, or over cable, or just be a reasonable file size.

Simply put, using a codec, you encode the Blender animation data to a video file which suits a particular output media such as PAL TV or NTSC TV. When you have used the encoded data to create a video CD or DVD, the CD or DVD is played in a device (CD / DVD Player) which decodes the data for display i.e. Television Screen.

As previously stated you must have the codec installed on your computer. Codec Packs are available for download from the Internet. Two examples are:

K-Lite Codec Pack 12.4.7
media.player.codec.pack.v4.4.2.setup.exe

19.9 Making a Movie

In the preceding information the procedure for rendering an image or a animation sequence has been briefly explained. In rendering the animation you first created a series of image files and then repeated the process creating a video file. The video file does not constitute a movie. In this case the video was a mere 10 frames, a Video Clip, but even if it were a thousand frames it would not be a movie. It is merely a render of one animation sequence from one Scene into a Video File. Movies are made by combining many video files and then rendering the combination to a Movie File. At the same time as this

combination is compiled sound effects are added and synchronised with the video.

This combining, synchronising and editing takes place in a **Video Sequence Editor** (VSE). Blender has its own VSE which is discussed in Chapter 29.

A Video Clip (movie file) will take some time to Render (compile) depending on the length of the animation. Each Frame of the animation has to be rendered and saved. Depending on the complexity of the Scene, a Frame can take from a few seconds to several minutes to render. To begin, it is best to keep everything very basic and simple. If you get to the stage where you have created a wonderful movie, you can send the animation files to a **Render Farm** on the Internet to have them rendered–it saves you time but it costs you money.

20 Animatior

20.1 Animation in the 3D Viewport
20.2 Movement in the 3D View Editor
20.3 Planning the Animation
20.4 Keyframes Time and Interpolation
20.5 Animation Speed and Length
20.6 Repositioning the Timeline Cursor
20.7 Inserting Keyframes
20.8 Scrubbing the Animation
20.9 Playing the Animation
20.10 Adding Keyframes
20.11 Automatic Keyframing
20.12 Controlling the Animation
20.13 The Dope Sheet Editor
20.14 The Graph Editor
20.15 Graph Editor Components
20.16 Selecting the Curve in the GE
20.17 Rotation - Euler
20.18 Graph Editor Cursor
20.19 Editing the Curve in the GE
20.20 Editing the Curve - Dope Sheet
20.21 Scaling the Dope Sheet
20.22 Other Types of Curves
20.23 The Curve Properties Panel
20.24 Animating Rotation

20.25 Rotating Using F-Curves
20.26 Animating Other Features
20.27 Keying Sets
20.28 Animation Follow Path
20.29 Evaluation Time
20.30 Keyframing
20.31 Displacement Sound Animation
20.32 Sound Effect and Cast Modifier
20.33 The Animation Workspace

Animatior is the illusion of motion, of making objects depicted in a still image appear to move. In its simplest form stick figures drawn on separate pages appear to move when the pages are viewed in quick succession. Animation has advanced from that simple technique to sophisticated full length feature films with sound and voice which are experienced today. In the past Movies were produced by laboriously drawing many images, posing figures, each slightly different to the next which were photographed and transcribed to film. Each image was then said to be a Frame in an animation sequence. Today the process is accomplished by Computer Graphics which essentially mimics that same process. In Blender animation is accomplished by creating data which displays on the computer Screen. The display is programmed to change over a period of time **(The Timeline)** and then captured by a camera at intervals producing **Frames** of the animation. Each Frame is **Rendered,** which means the Blender data is correlated and turned into a series of digital images. The images are compiled into a video clip or sequence depicting an action. Finally a series of video clips are assembled, edited, combined with effects such as audio and converted to a movie file.

This chapter will explore some of the techniques used for creating animation effects.

20.1 Animation in the 3D Viewport

Animation may be performed in the default Blender Screen, in the 3D Viewport Editor, in conjunction with the Timeline Editor. Alternatively Blender has a dedicated **Animation Workspace (**see 20.33).

To demonstrate the very basics of the animation process the default **3D Viewport Editor and Timeline Editor** will be used. The Animation Workspace will be introduced at a later stage.

In the 3D Viewport Editor a **Cube Object** will be animated to move in the Scene, and change shape at the same time. Simple motion and deformation of an Object are only two of many features which may be animated.

The objective in creating an animation is to capture what you see on the Screen in the **Camera View.** This will be what is included in the final render. You may set up an animations sequence in the Blender Scene then position the Camera or position multiple Cameras to capture parts of the sequence. What is captured will be rendered to a series of still images (Frames) producing a video clip. The clip will be finally compiled (spliced together) with other clips to make a movie file.

20.2 Movement in the 3D View Editor

Moving or Translating Objects in the 3D Viewport Editor may be performed by selecting the Object, pressing the **G Key** and moving the Mouse or by using the **Manipulation Widget.** The two methods move the Object in the Scene in different ways.

What you see in the 3D Viewport Editor when you move Objects depends on the method of Translation. In Figure 20.1 **Cube A** is Translated by pressing the G Key (Grab) and moved up and to the left. **Cube B** is translated using the

Move Tool which activates the Manipulation **Widget** (press the T Key to see the Tool Panel).

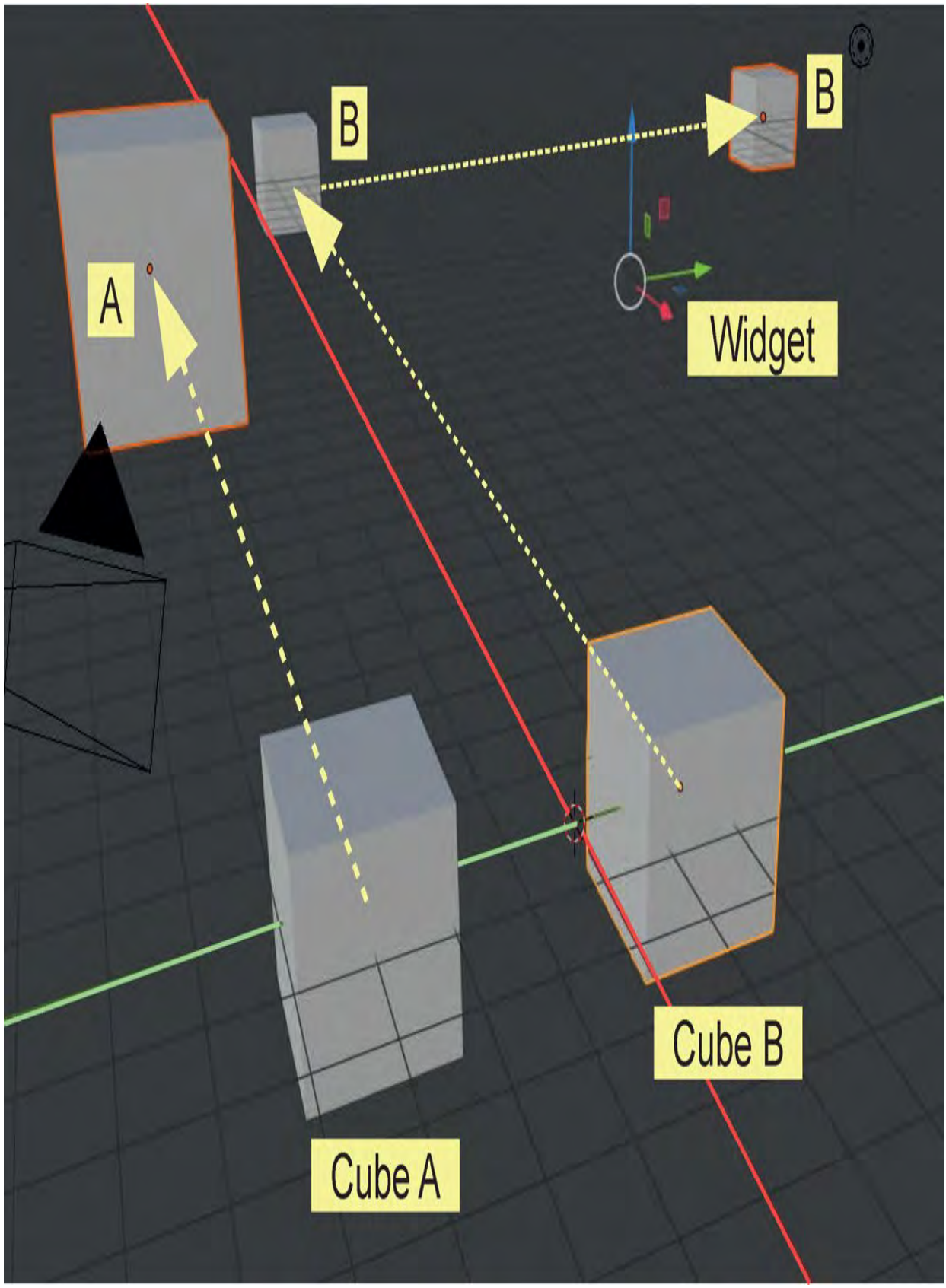

B
B
A
Widget
Cube B
Cube A

Figure 20.1

Note: The Widget in the diagram is positioned at the center of geometry of the three B Cube Objects. This is for diagram construction only.
When **Cube A** is Translated by using the **G Key** the movement is confined to the plane of the computer Screen.

When **Cube B** is Translated by using the Move Tool Widget the movement is confined to either the X, Y or Z Axis of the 3D World (the Scene).

Cube B has been moved back along the X Axis then to the right along the Y Axis. In **User Perspective View** you see its size diminish as it recedes into the distance. In fact, all views of the Cube have the same physical size as seen in Top Orthographic View (Figure 20.2).

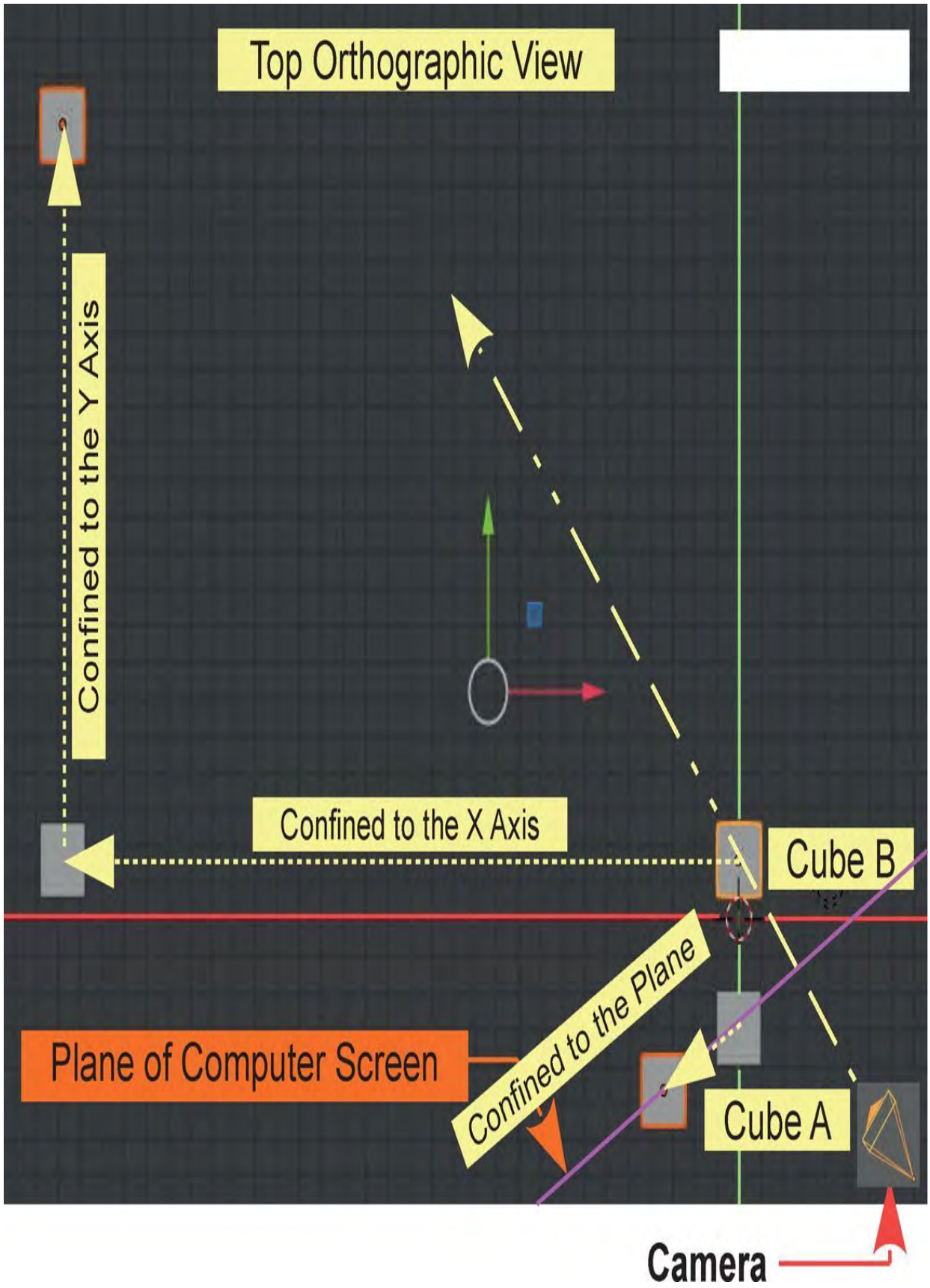

Top Orthographic View
Confined to the Y Axis
Confined to the X Axis
Cube B
Confined to the Plane
Plane of Computer Screen
Cube A
Camera

Figure 20.2

Understanding this concept of Translation on the Computer Screen and in the 3D World will assist when animating Objects to move. Screen and in the 3D World will

20.3 Planning the Animation

Planning the animation is an important part of the process. Without having even a rudimentary idea of what you want to achieve can lead to disaster when the Scene becomes a little complicated.

In demonstrating the concept of animation, with a view to illustration, begin with a very simple sequence. The default Cube Object in the default 3D Viewport Editor will be relocated (Translated) then animated to move forward along the X Axis of the 3D World and at the same time reduced in size and Scaled along the Y Axis.

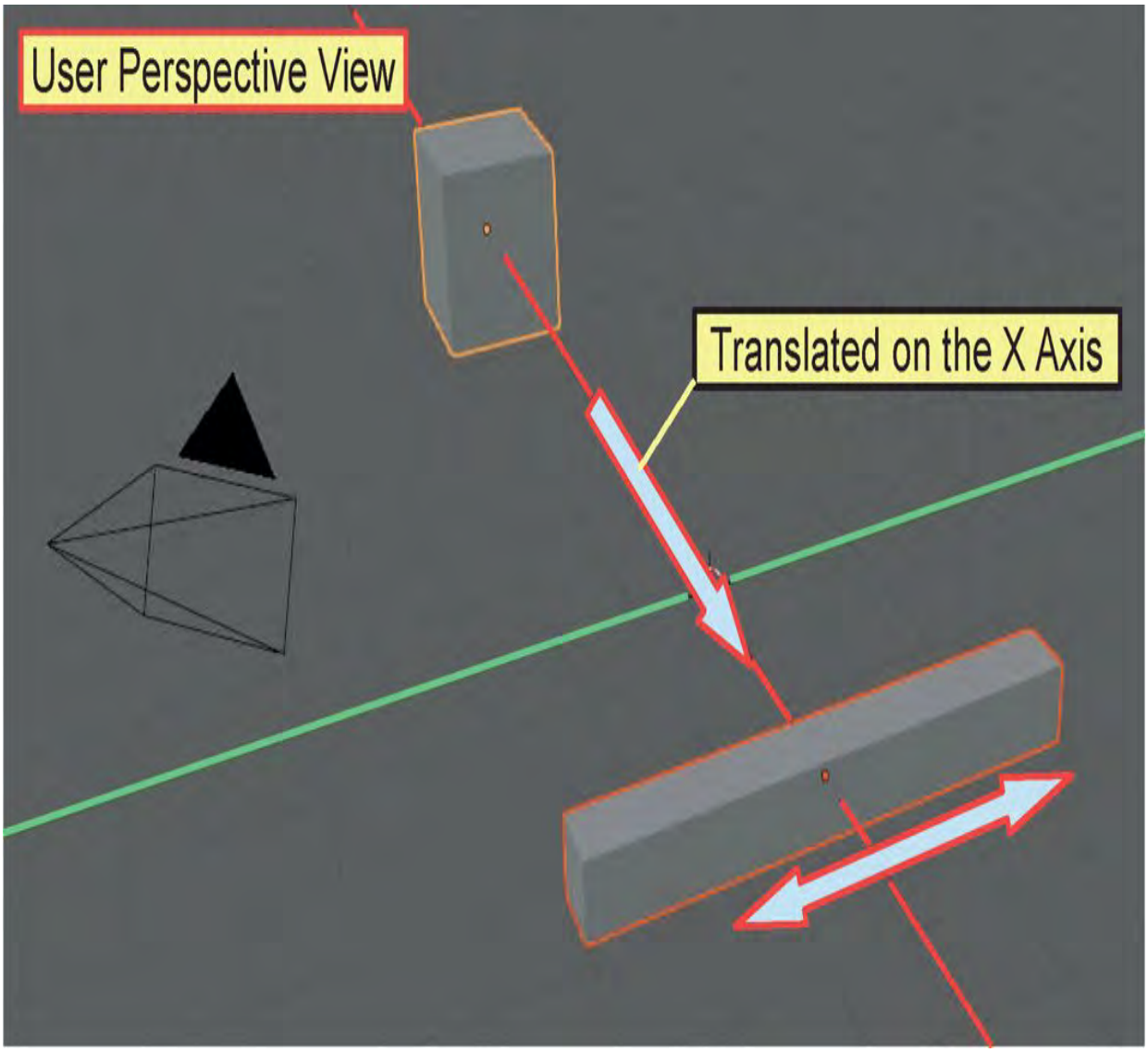


Figure 20.3

Figures 20.3 and 20.4 show the default Cube Object positioned approximately ten Blender units on the X Axis towards the back of the Scene in the 3D Viewport Editor (minus 10 Units).

The Cube will be animated to move forward along the X Axis.

At the same time the Cube will be Scaled on the X, Y and Z Axis.

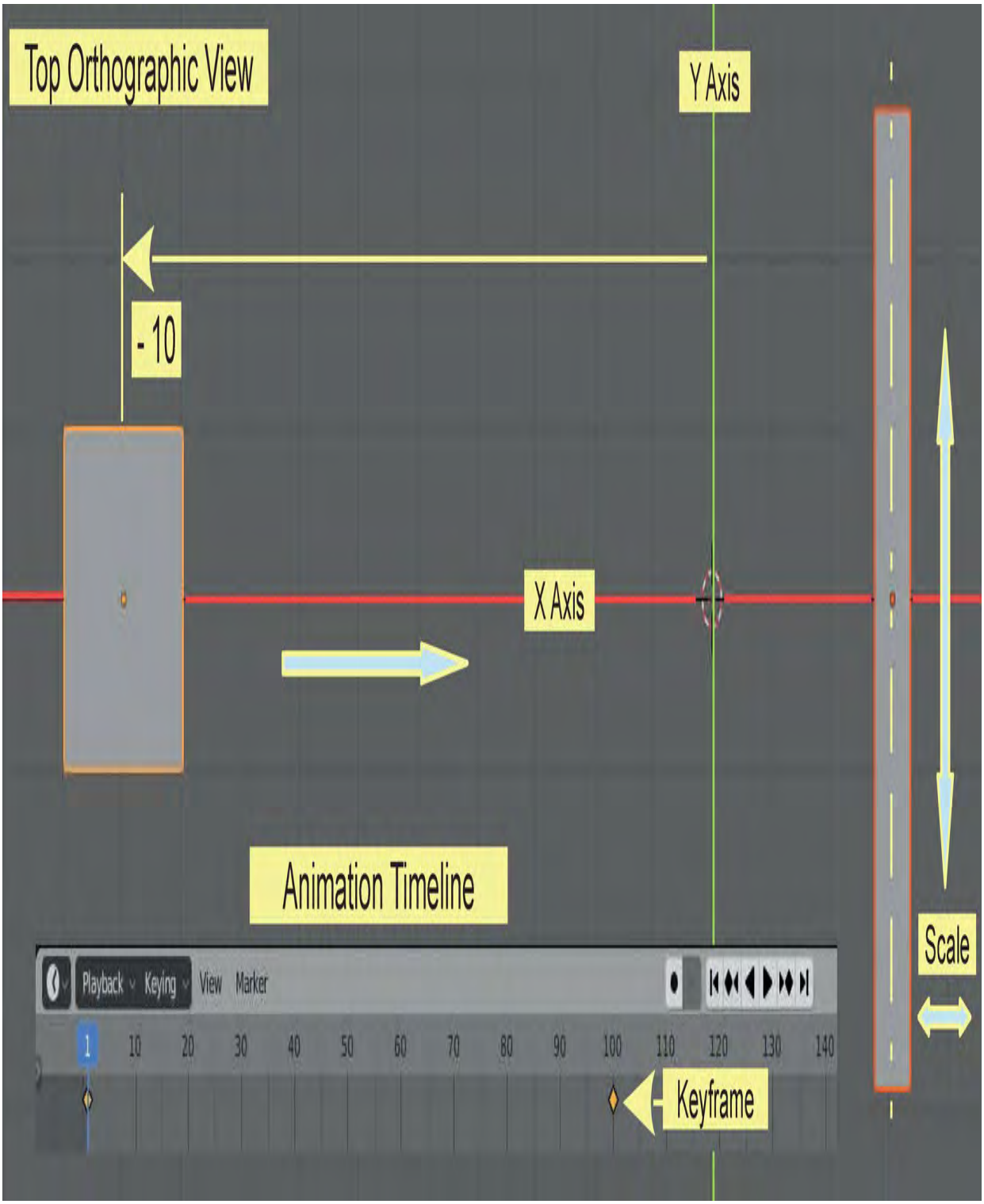


Figure 20.4

20.4 Keyframes Time and Interpolatior

The first step in an animation is to decide what you want your actor to do in a given time. In this demonstration, the actor will be the Cube Object. How long it takes the actor to do something will depend on how many Frames per second the animation is run and this is determined by which format your final render will be.

The **Render Format** determines how many Frames per second the animation should run (For example when playing in a television format, **NTSC** for the US at 30 fps, **PAL** for Australia at 25 fps). When considering the animation, make the motion occur in an appropriate time. Look at the frames per second and relate it to time. If you want a movement to take 3 seconds and you are running at 25 frames per second, then the animation has to occur in 75 Frames (3 x 25 = 75).

In Blender you do not have to create every single Frame of the animation. You set up single Frames **(Keyframes)** at specific points and the program works out all the intermediate Frames.

Think of a 10-second animation that, when running at 25 Frames per second, would consist of 250 Frames. If you want your actor to go from point A to point B and then to point C in the Scene within the 250-Frame animation, you first insert a Keyframe at Frame 1 with the actor at position A. This is giving Blender data that says, at the Frame 1, locate the actor at location A. Then at another Frame, mid way in the animation, insert a second Keyframe with the actor at location B. Finally insert a third Keyframe at frame 250 with the actor at location C. These are the Keyframes for the animation. Blender will work out all the in-between Frames. The Keyframes can also include the data for other features such as scale, rotation and color.

Determining the in-between data is called **Interpolation.** There are different methods of interpolation. By default, Blender uses **Bezier Interpolation,** which for motion gives a nice acceleration and deceleration between **Keyframes.** When an object moves from point A to point B in a

given time, it is said to move at a certain velocity (speed). In theory, the speed could be represented as a straight line graph, but in practise an Object at rest (motionless) has to go from being motionless to moving at a certain velocity. The rate at which it attains the velocity is called acceleration. Blender's **Bezier Interpolation** draws curves at the beginning and end of the straight line graph (acceleration and deceleration). You have the options to choose **Constant** or **Linear** type interpolation if appropriate. Selection of interpolation types will be discussed later in the chapter.

Using the term **Bezier** to describe interpolation is in fact an anomaly. Bezier actually describes a type of line (the line on a graph described in the previous paragraph). A Bezier line or curve in Blender is a line that has control points that allow the shape of the line to be altered or edited (see Editing Using Curves Chapter 10). In Blender, the control points are located at the position of the **Keyframes.** Interpolation is performed according to a mathematical formula that determines the shape of the line. When the data for the Frames in the animation is drawn as a line on a graph, the line conforms to that mathematical formula. For the moment, accept the default Bezier-Type interpolation to demonstrate the insertion of Keyframes and the creation of a simple animation.

20.5 Animation Speed and Length

Note: Frame Rate 24fps in the **Dimensions Tab.**

The **Timeline Editor** Frame range settings are, **Start: 1** and **End: 250** (Figure 20.6). This is saying the animation will begin at Frame 1 and end at Frame 250. Running at the rate of 24 Frames per second will give an animation time of approximately 10 seconds. If you think about it, 10 seconds is quite a long time for a single action to take place in a video clip. Also in the Timeline Editor, make note of the lighter grayed area beginning at Frame 1 and ending at Frame 250. Changing the **Start Frame** and **End Frame** values

in the header panel will move the end positions of the lighter grayed area. The vertical blue line at Frame 1 is the **Timeline Cursor.**

The default 24 frames per second, suitable for PAL format, as set in the **Properties Editor, Output Properties** will be used.

Figure 20.5

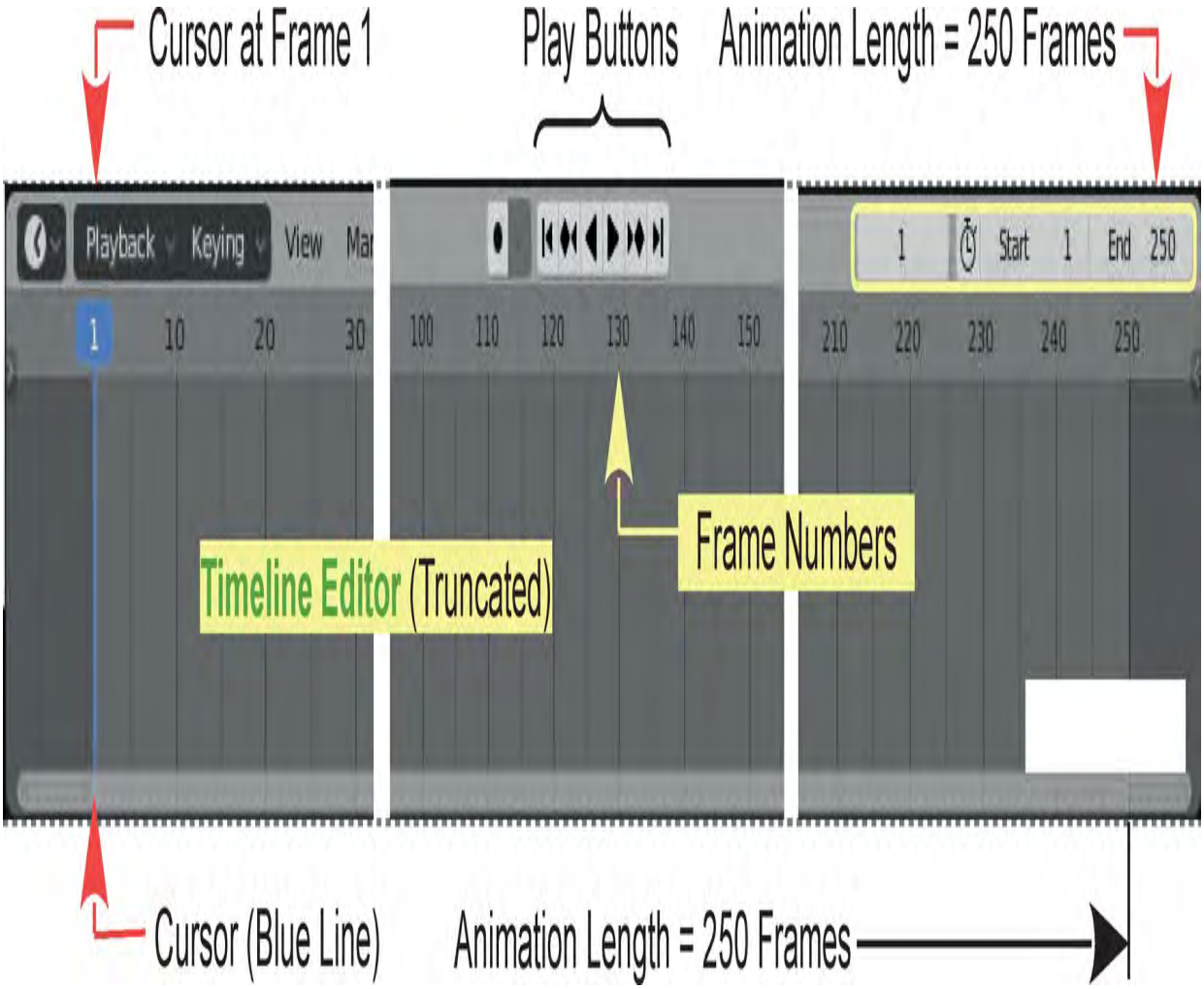


Figure 20.6

To make the process relatively simple and suitable for a demonstration, the actor (the Cube) will be made to move in a straight line along the X Axis and at the same time increase in size on the Y Axis. Make sure the Cube is selected in the 3D Viewport Editor. Initially only two **Keyframes** will be inserted.

In the default Scene, the actor (the selected Object — the Cube) is located at Frame 1 in the animation. In the upper LH corner of the 3D Editor, you will see **(1) Collection I Cube** in white lettering. This indicates that you have the

Cube selected at Frame 1 in the Animation Timeline. If you had ten objects in the Scene, all of which were actors with perhaps some hidden, it's nice to know which one is selected and where they are in the Timeline.

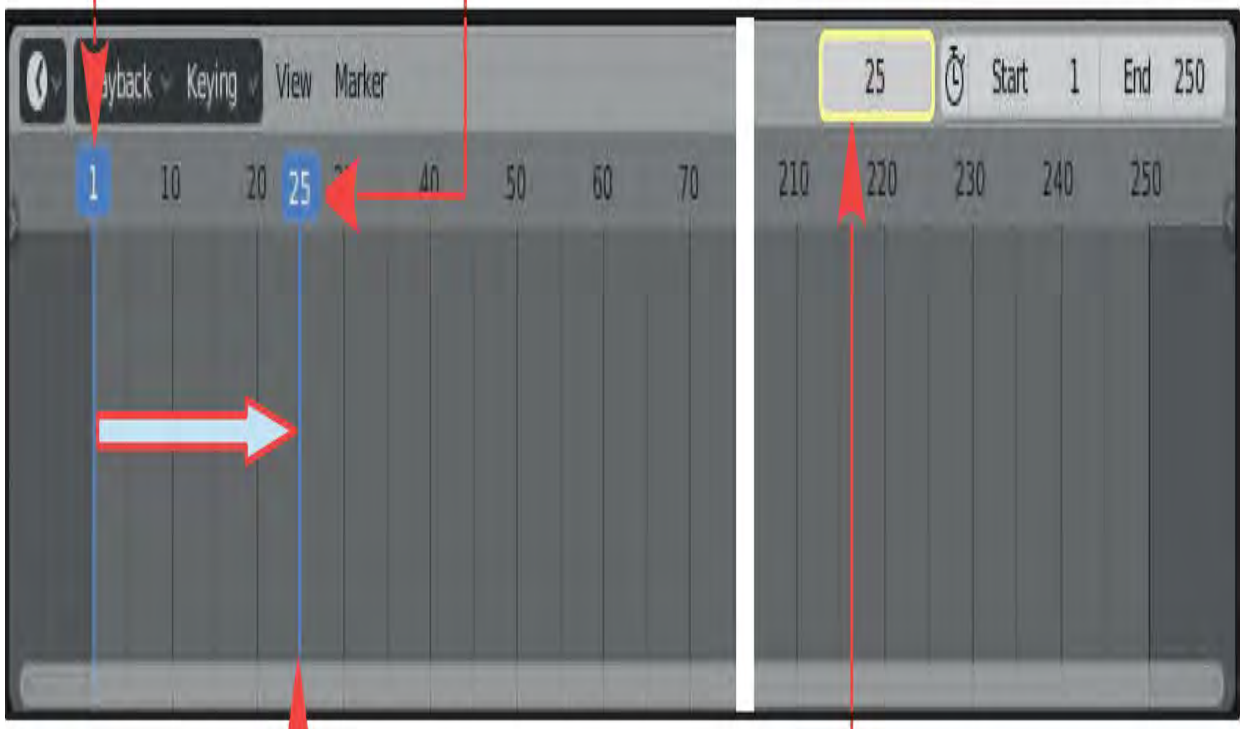
20.6 Repositioning the Timeline Cursor

In the **Timeline Editor** at the bottom of the Screen the buttons labelled **25, Start: 1 and End: 250,** show the current Frame, the Start Frame and the End frame set for default animation (25 displays when the Cursor is relocated at frame 25).

Clicking LMB on the scale repositions the Cursor (blue button with blue line). With the mouse cursor in the **Timeline Editor** pressing **Num Pad + or —** zooms the scale. Scrolling the Mouse Wheel zooms the scale.

In the default view the **Timeline Cursor** (blue button with vertical blue) line is located at **Frame 1.** Click on the cursor (blue button) with the LMB, hold and drag it across to Frame 25 (Figure 20.7). Note the number in the blue button changes and also next to **Cube** at the upper LH side of the 3D Viewport Editor and in the Header bar of the **Timeline Editor.** Other ways to change the Frame are to click on the little arrows on either end of the **Frame Number** in the Timeline Editor Header, or click LMB on the button, hold and drag to change the Frame number or click on the **button,** hit delete, and retype the required frame number.

Click, Hold and Drag or Click RMB in the Editor at a Frame to reposition the Cursor.



Cursor at Frame 25

To accurately locate the Timeline Cursor
Position Mouse Cursor over the button.

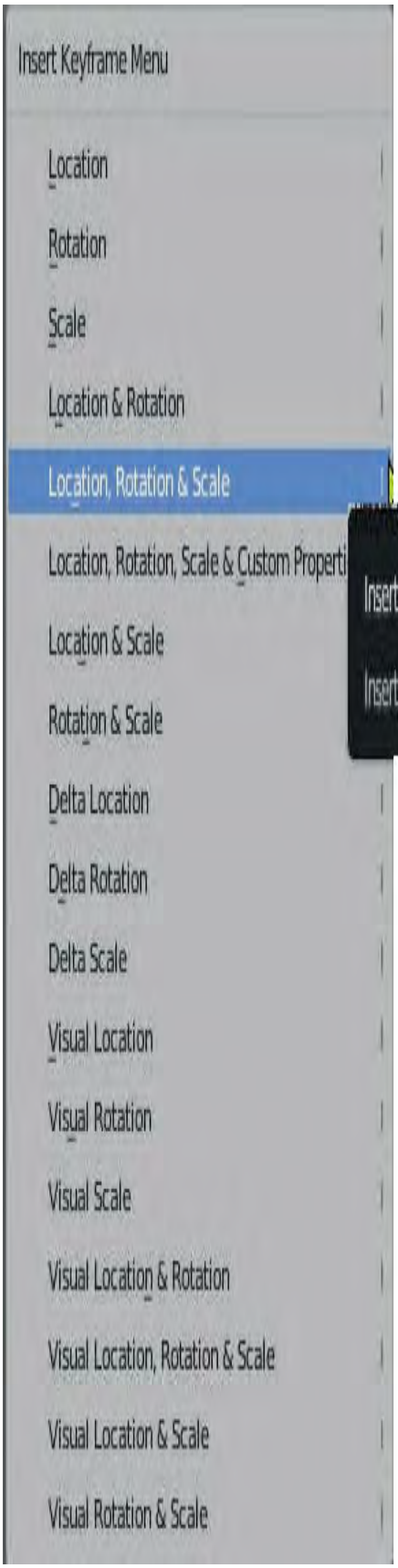
Drag Left or Right
Click to increment the Frame
Double Click – Delete – Retype a Frame Number

Figure 20.7

20.7 Inserting Keyframes

Keyframes are inserted in the animation at specific Frames. For example, with the Cube Object located at minus three (-10) units on the X Axis, position the **Timeline Cursor** at Frame 25.

Place the **Mouse Cursor** in the **3D Viewport Editor** and press the **I key** to insert a **Keyframe.** In the selection list that displays, select (click) **Location,Rotation & Scale,** which covers and the size of the

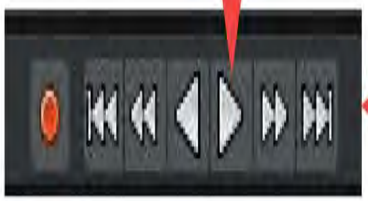


You must have the Cube selected. You will see an **orange diamond** appear at Frame 25, in the Timeline Editor.

Figure 20.8

At this point, only one Keyframe has been inserted and the Cube remains stationary at minus ten units on the X Axis of the Scene.

If you click LMB on the **Timeline Editor Cursor** (blue button) ,hold and drag the mouse from Frame 25 along the timeline the Cube (actor) remains stationary. Clicking LMB and dragging the blue line in the Editor is called **scrubbing the animation**, which is actually manually playing the animation. You can play the animation by clicking the **Play button** in the **Timeline Editor Header**.



Note: With the Timeline Editor Cursor positioned at a Keyframe the notation in the 3D Viewport Editor indicating the selected Object changes color.

Note: Going from Frame 1 to Frame 25 at 24 Frames per Second equals approximatly 1 second.

Figure 20.9

Remember; There is only one **Keyframe** which tells Blender that, at Frame 25 the Cube Object is positioned -10 units in the Scene. There is no other information, therefore, playing the Animation does nothing.

Continue creating the animation by moving the Cursor to Frame 100 (click on Frame 100).

In the 3D Viewport Editor, grab and position the Cube +3 Blender units on the X Axis (-10 to +3). Scale the Cube down and Scale up four on the Y Axis (Figure 20.10). With the Mouse Cursor in the 3D View Editor, press the **I key** and select **Location,Rotation & Scale** to insert a second at Frame 100. You will see another set of diamonds in the Timeline at Frame 100 (Figure 20.12).

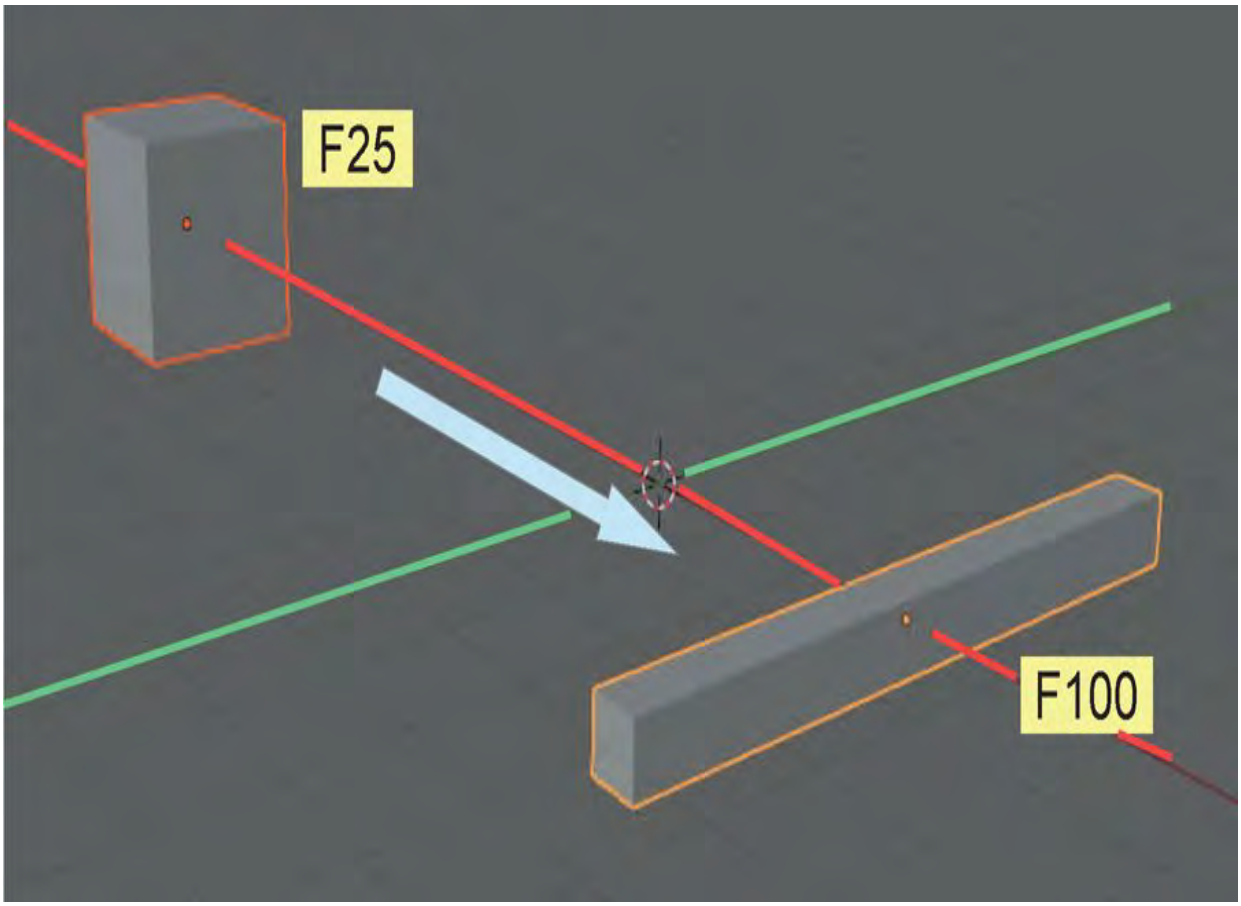


Figure 20.10

To accurately locate and scale the Cube, with the Mouse Cursor in the 3D View Editor, press the **N Key** and enter values in the **Object Properties** panel (Figure 20.11).

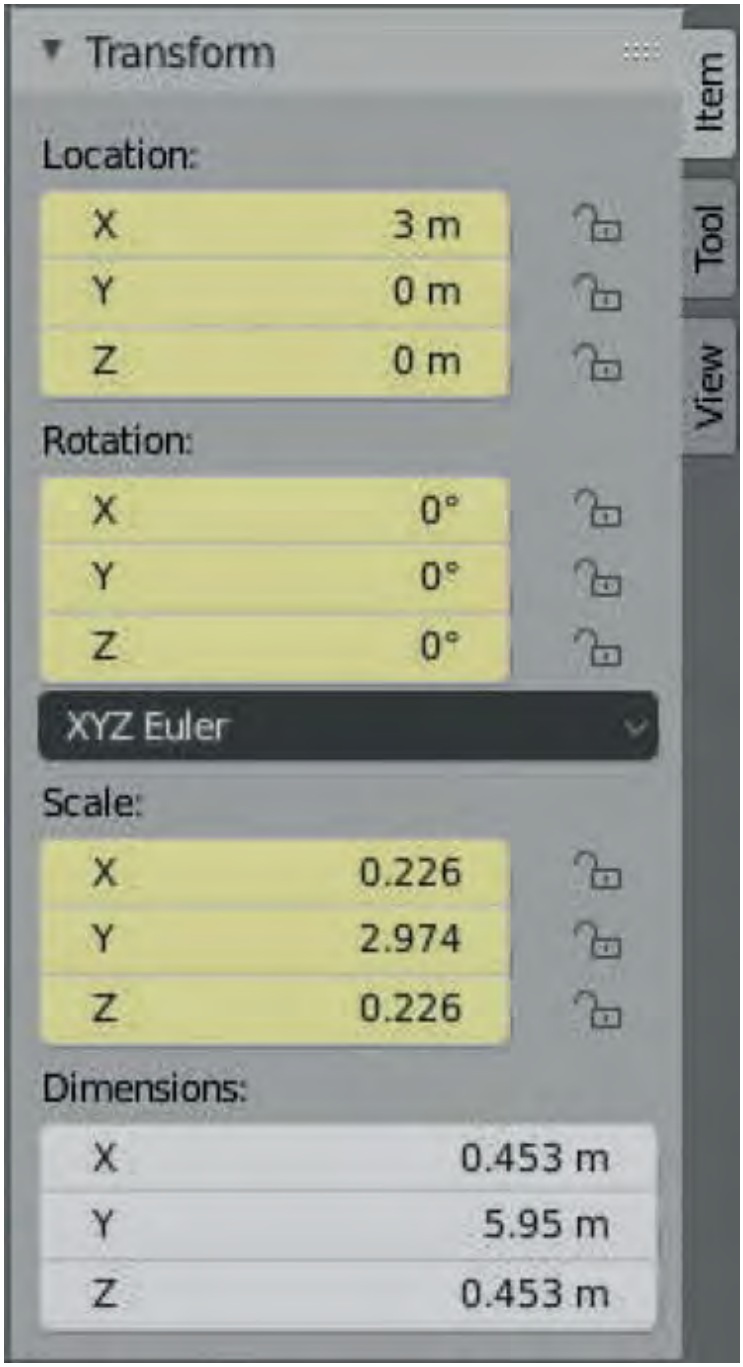


Figure 20.11

Values shown are for Location & Scale at Frame 100.

The diamonds in the Timeline Editor merely show that the Frame has been set as a Keyframe. They do not indicate the status which is required to set the Animation to include the Location, Rotation and Scale properties which result in the change occurring between Frame 1 and Frame 100 as displayed in Figure 20.12. To see a more comprehensive view of status you need to expand the display.

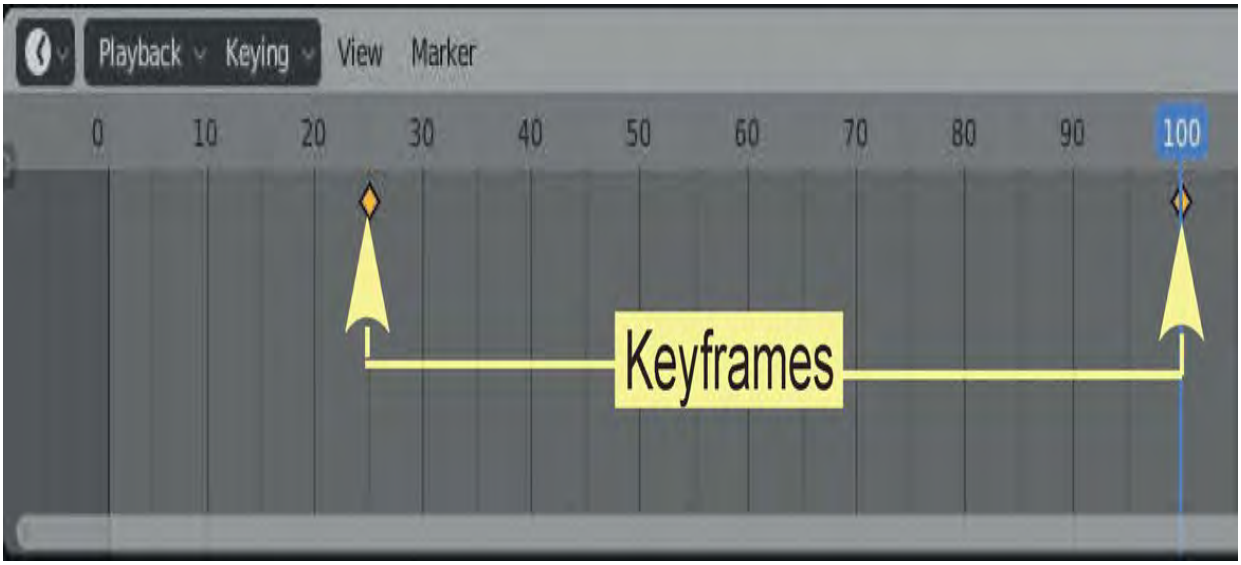


Figure 20.12

N Key to hide the panel.

In the upper left hand corner of the **Timeline Editor** mouse over on the little arrow just below the Editor Icon.

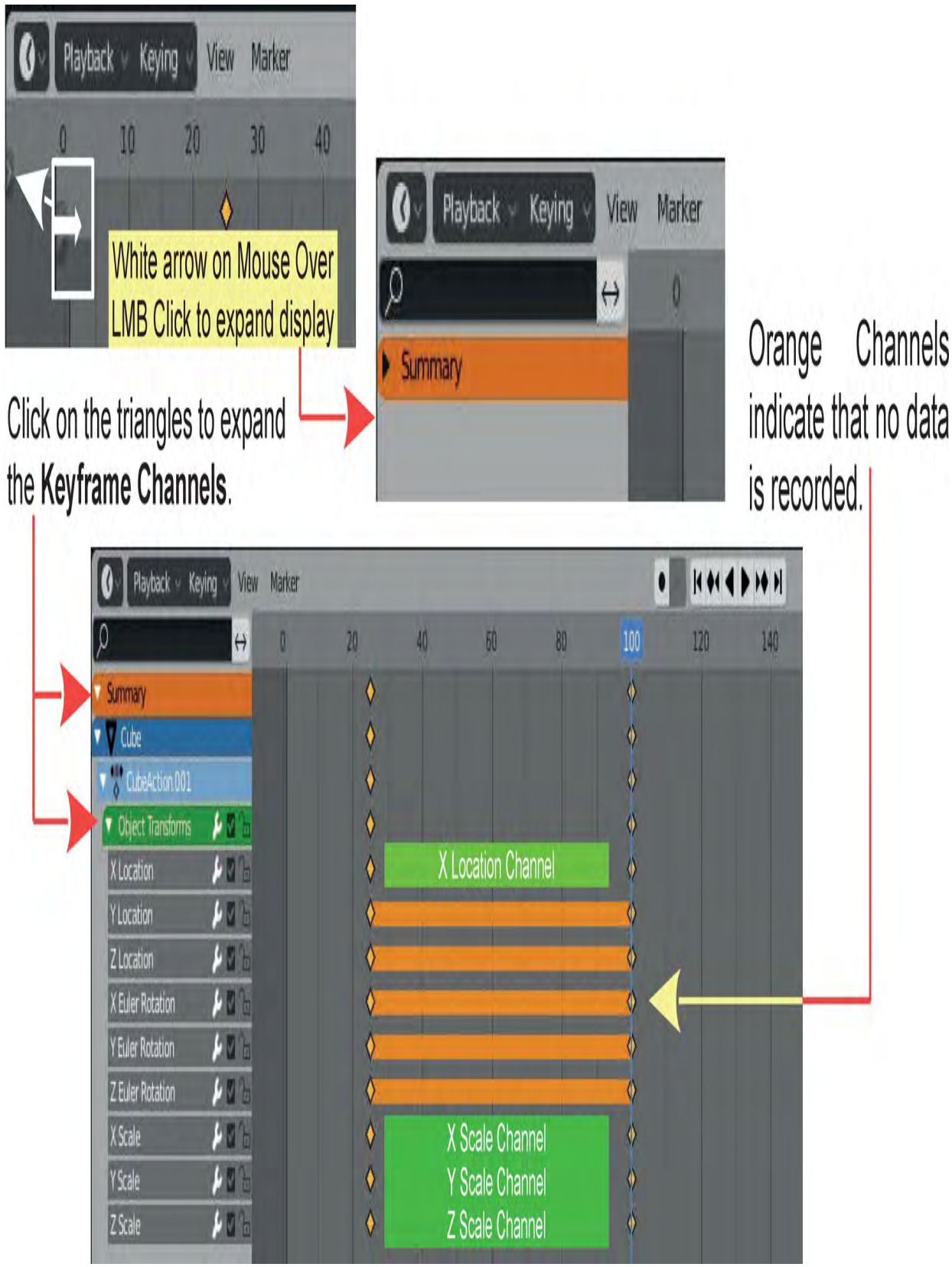

Playback
Keying
View
Marker
0
10
20
30
40
White arrow on Mouse Over
LMB Click to expand display
Summary
Click on the triangles to expand the Keyframe Channels.
Orange Channels indicate that no data is recorded.
0
20
40
60
80
100
120
140
Cube
CubeAction.001
Object Transforms
X Location
Y Location
Z Location
X Euler Rotation
Y Euler Rotation
Z Euler Rotation
X Scale
Y Scale
Z Scale
X Location Channel
X Scale Channel
Y Scale Channel
Z Scale Channel

Figure 20.13

Keyframe Channels display for each dataset Animated, in this case Location, Rotation and Scale (Green Channels). **Rotation** is included since **Location, Rotation & Scale** was selected when entering the Keyframes but no data has been recorded (Orange Channels). The Cube was not actually Rotated in the Animation neither were Y and Z Location.

20.8 Scrubbing the Animatioi

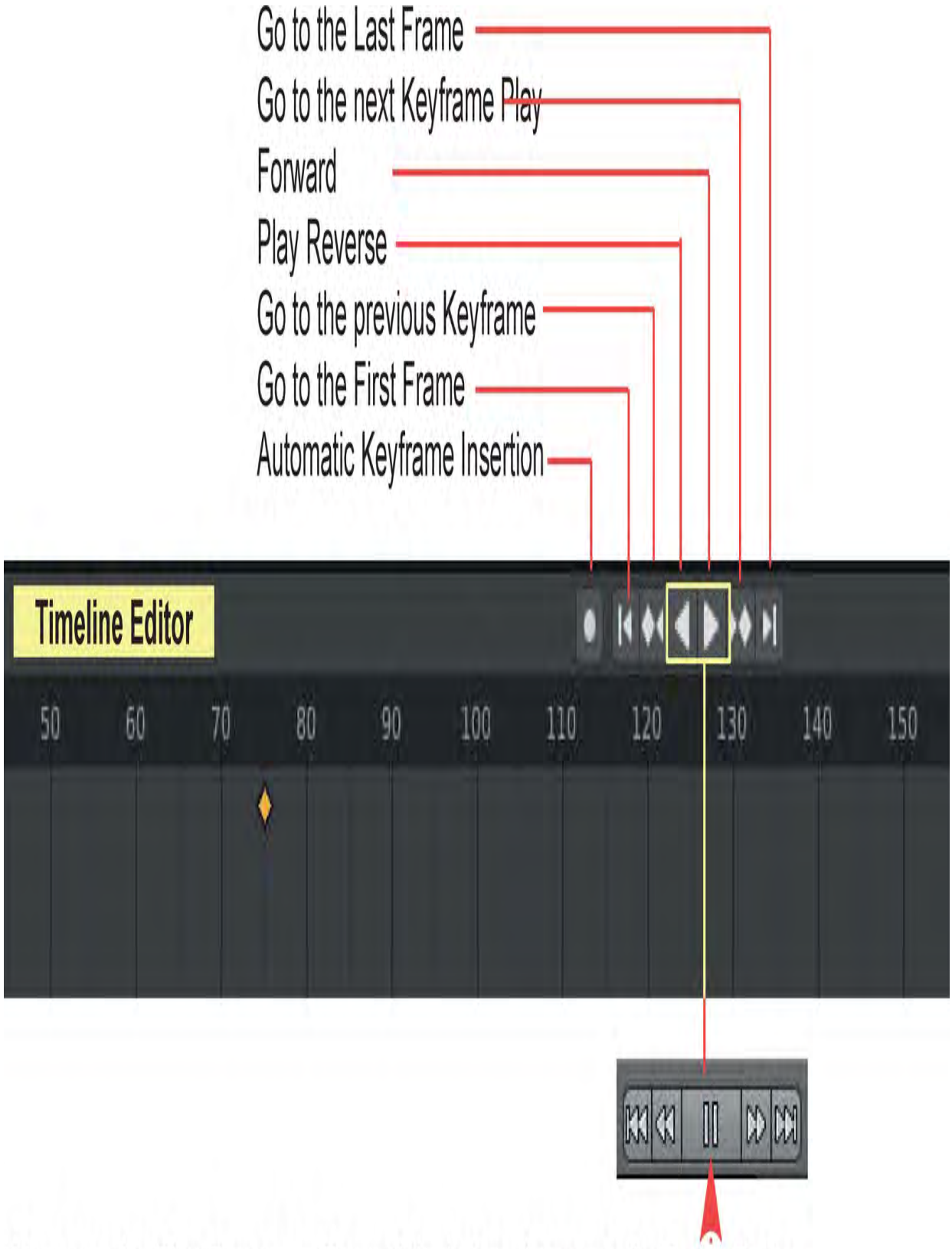
When you **scrub the animation** (Click, Hold and Drag the Blue Line Cursor, in the Timeline Editor or in the Dope Sheet Editor) between Frames 25 and 100, you will see the Cube move and change in size—you are manually playing the animation; **Scrubbing.** Note that the action only takes place between frames 25 and 100, which are the location of the Keyframes; no action takes place on either side of the Keyframes.

Note: In moving the Cube from minus ten to plus three on the X Axis the Cube will be disappearing from Camera View. This may require addressing since Camera View is what Renders as the final animation.

20.9 Playing the Animation

To actually play a **preview of the animation,** move the Cursor in the Timeline to Frame 1 then press the **Spacebar** with the Cursor in the 3D Viewport Editor. Say "one thousand" to yourself slowly (counting one second, while the Cursor in the **Timeline** moves across to frame 25). You will see the Cube remain stationary until the Cursor reaches frame 25 then the Cube will move and change in size. At frame 100, it stops moving and changing size. The Cursor in the Timeline continues on to frame 250 then jumps back to frame 1 and the preview of the animation plays again. Press **Esc** or press the Spacebar a second time to stop playing.

Another way to play the animation is to press the **Play** buttons in the **Timeline Editor Header** (Figure 20.14). These button is much like the play button on any video or audio player.



When pressing Play, Reverse or Forward the button changes to the **Pause Button**.

Figure 20.14

20.10 Adding Keyframes

Keyframes may be added to the animation to translate (move), scale and rotate the actor in the Scene. For the most part, location and size Keys work flawlessly but care needs to be taken with Rotation Keys . If you try to rotate an Object too far in one set of Keys, the Object may not rotate in the direction you want it to and it may rotate oddly. Try small angular movements between Keys while rotating. There are better ways to control this and tools to simplify the process, (see 20.14). Be aware that the movement of the actor may not be exactly as planned. Blender automatically defaults to trying to create a smooth flow through the animation.

To insert additional Keyframes, position the Timeline Cursor, adjust the status of the Actor then press the I Key.

20.11 Automatic Keyframing

Keyframes have been inserted in the animation by placing the Mouse Cursor in the 3D Viewport Editor, moving to a Frame in the Timeline, changing the status of the Object and then pressing the **I Key** and selecting one of the Keyframe options.

Blender has an **Automatic Keyframe** insertion function which is activated by pressing the white button in the **Timeline Editor Header** (highlights blue when active)(Figure 20.15).

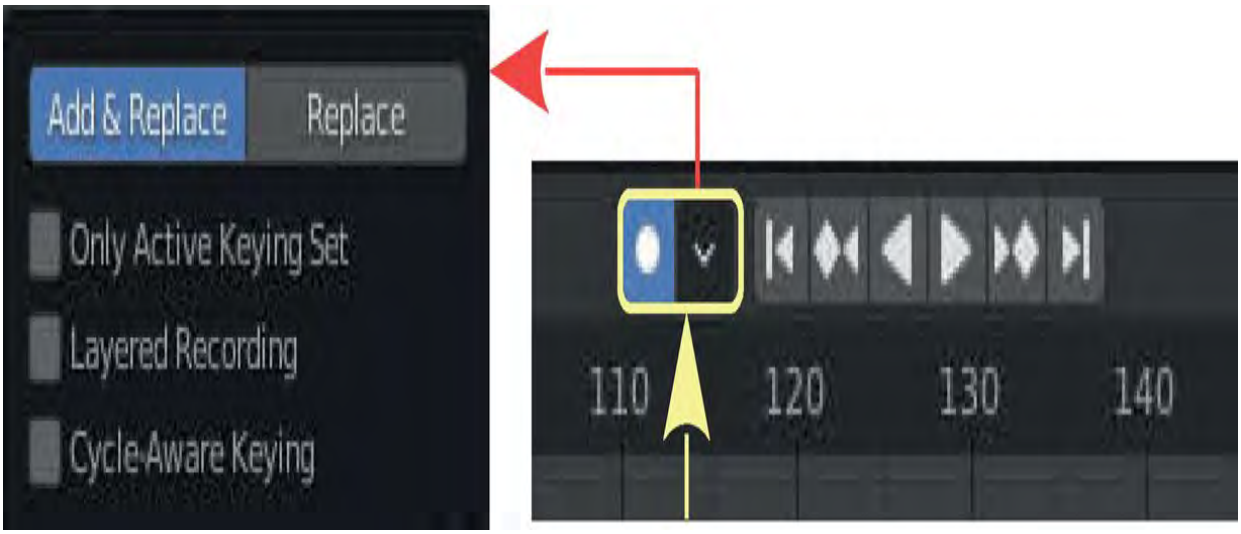


Figure 20.15

With auto on, whenever you Move, Scale, or Rotate the actor Object in the 3D Viewport Editor, a Keyframe will be inserted at whatever Frame has been selected in the Timeline.

For example; With the Cube Object in the default Scene in the 3D Viewport Editor it is located at the intersection of the X Axis and Y Axis. The Cursor in the Timeline Editor is located at Frame 1. With Auto on, Translating the Cube along the X Axis inserts a Keyframe at Frame 1 recording that this is the state of the Cube at Frame 1.

If the Timeline Cursor had been positioned at Frame 25 then the Cube Translated, a Keyframe will have been inserted at Frame 25. This will be the First Frame in the animation. If the Timeline Cursor is moved back to Frame 1 the Cube remains in its position for frame 25.

The procedure for Automatic Keyframing is, position the Timeline Cursor, Translate, Rotate or Scale the selected Object to insert Keyframes.

Remember to turn this off after you're finished using it (press the button a second time).

20.12 Controlling the Animation(

When an animation has been created it may be controlled (modified / adjusted) to fine tune the action perhaps to create an interaction with other Actors which are animated in the same Scene.

Controlling the Animation the Animation can be achieved by making adjustments in the 3D Viewport Editor and or in the Timeline Editor. To demonstrate the rudimentary concept of fine tuning continue with the Animation depicted in Figure 20.10.

The Cube Object moves from minus ten Blender units on the X Axis to plus three Blender units and changes shape as it moves. An adjustment will be made to delay the Y Axis Scale extension until Frame 160 in the Timeline. This means the Cube will come to rest at Frame 100 but will continue to extend on the YAxis until frame 160.

To achieve this delayed extension you could have the Cube selected at Frame 100, reduce the Scale on the Y Axis then Press the I Key and select Location, Rotation & Scale again which redefines the Cube's status at Frame 100. Reposition the Timeline Cursor at Frame 160, Scale the Cube up on the YAxis and insert new Location, Rotation & Scale Keyframes.

Figure 20.17

20.13 The Dope Sheet Editor

The **Dope Sheet Editor** introduces yet another means of Controlling the Animation. When comparing the Timeline Editor (Figure 20.16) to the Dope Sheet Editor (Figure 20.17) on the previous page they appear to be identical. Creating an Animation with Keyframes in the Timeline Editor then changing the Timeline to the Dope Sheet shows the same Keyframes and Channels. You

will note, however, that the Dope Sheet Editor has several more option buttons included in the Header. It is not intended to describe all the options and features of either the Timeline Editor or the Dope Sheet Editor, merely to introduce each, making you aware that they exist and are in many respects similar. To demonstrate one feature of the Dope Sheet, the extension of the Cube along the Y Axis of the Scene at Frame 160 may be controlled (adjusted) in the Editor.

A simpler method of making this adjustment is to relocate the Y Scale Keyframe in the Timeline Editor. With the Mouse Cursor in the Timeline, click LMB to deselect all the Keyframes. LMB click on the **Y Scale Keyframe** (orange) and drag the Mouse placing the **Keyframe at Frame 160**.

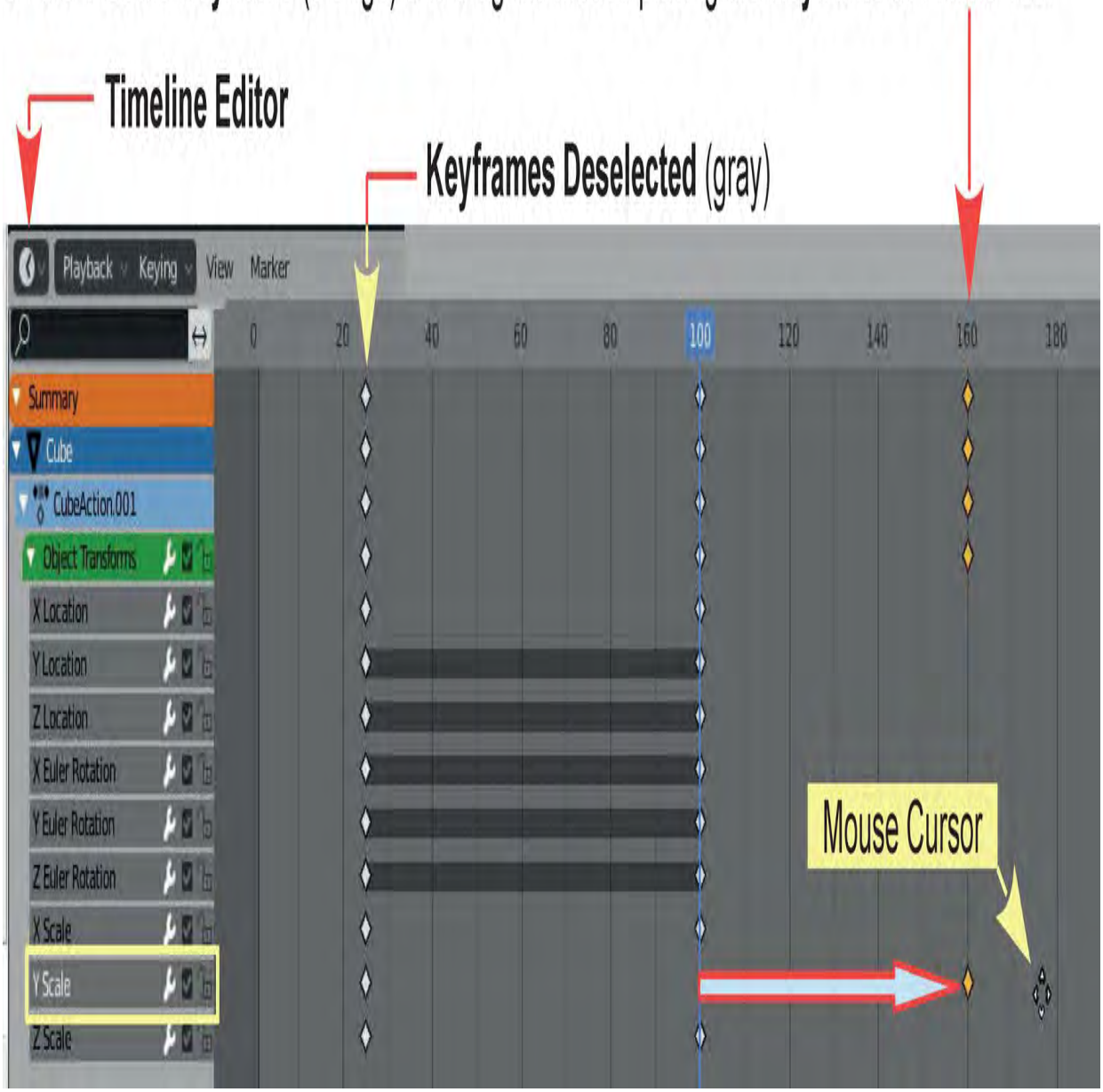


Figure 20.16

Clicking **View** in the Dope Sheet Editor Header reveals a selection of options (Figure 20.18). By checking **Show Sliders,** control Sliders are

displayed for each Animation Channel. In the diagram the Y Scale Keyframe has been relocated to Frame 160 per the previous exercise using the Timeline Editor and remains selected. The slider is highlighted yellow. By adjusting the value in the Slider you change the elongation of the Cube in the 3D Viewport Editor with the Cube at Frame 160.

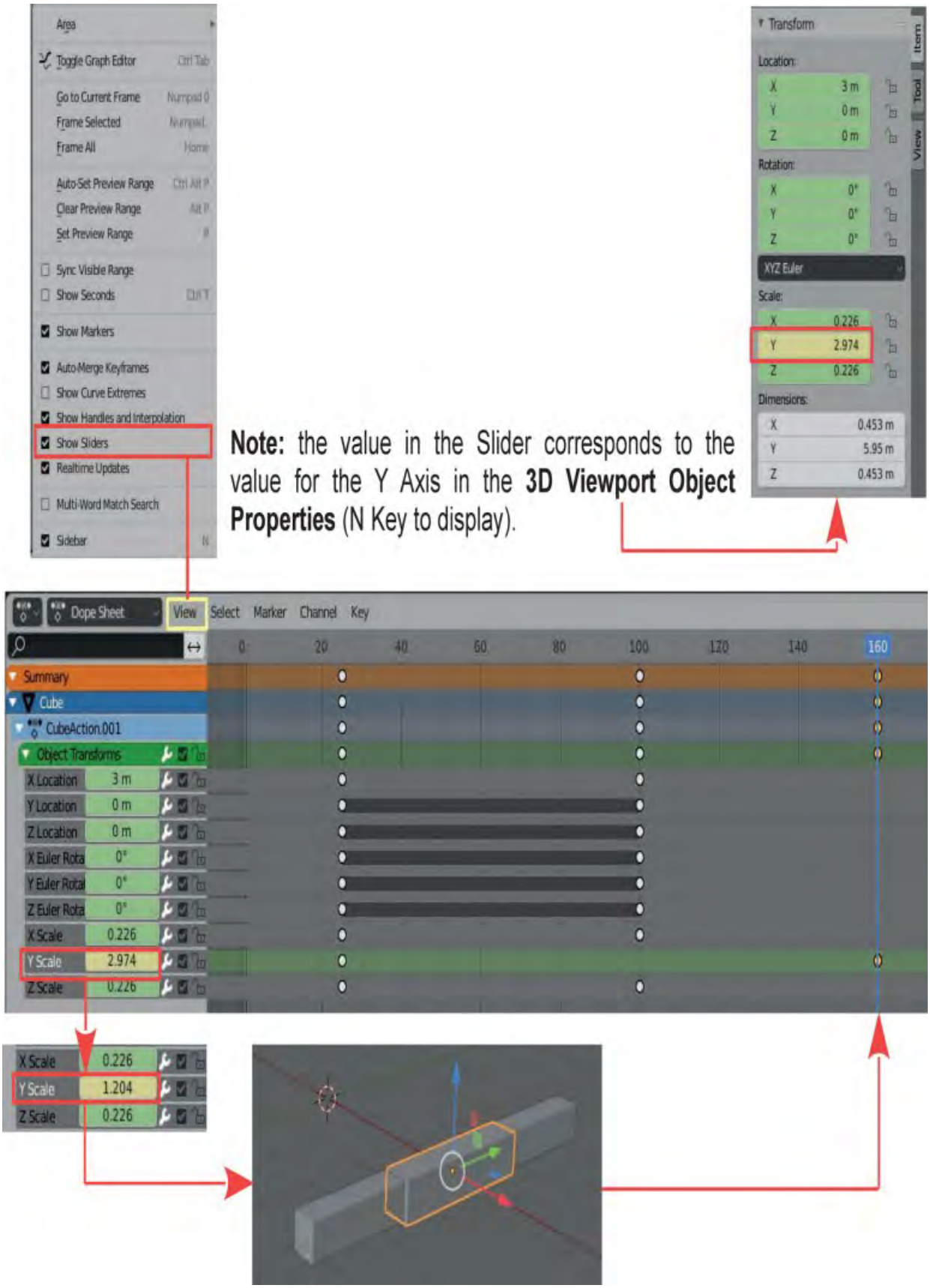

Area
Toggle Graph Editor Ctrl Tab
Go to Current Frame Numpad 0
Frame Selected Numpad .
Frame All Home
Auto-Set Preview Range Ctrl Alt P
Clear Preview Range Alt P
Set Preview Range P
Sync Visible Range
Show Seconds Ctrl T
Show Markers
Auto-Merge Keyframes
Show Curve Extremes
Show Handles and Interpolation
Show Sliders
Realtime Updates
Multi-Word Match Search
Sidebar N
Transform
Location:
X 3 m
Y 0 m
Z 0 m
Rotation:
X 0°
Y 0°
Z 0°
XYZ Euler
Scale:
X 0.226
Y 2.974
Z 0.226
Dimensions:
X 0.453 m
Y 5.95 m
Z 0.453 m
Item
Tool
View
Note: the value in the Slider corresponds to the value for the Y Axis in the 3D Viewport Object Properties (N Key to display).
Dope Sheet
View
Select
Marker
Channel
Key
0
20
40
60
80
100
120
140
160
Summary
Cube
CubeAction.001
Object Transforms
X Location 3 m
Y Location 0 m
Z Location 0 m
X Euler Rota 0°
Y Euler Rota 0°
Z Euler Rota 0°
X Scale 0.226
Y Scale 2.974
Z Scale 0.226
X Scale 0.226
Y Scale 1.204
Z Scale 0.226

Figure 20.18
Figure 20.19

Another way to control Animation is to employ the **Graph Editor.** Each of the Animation Channels in the Timeline Editor or Dope Sheet Editor may be displayed as a Graph Line.

20.14 The Graph Editor

The Graph Editor shows a graphical display of the animation. The graphs can be edited to refine and control the animation actions. In Figure 20.20 the 3D Viewport Editor has been changed to the Graph Editor with the Dope Sheet Editor below. The Viewports have been aligned by manipulating the scales, zooming and repositioning the Viewport to follow).

Graph Editor

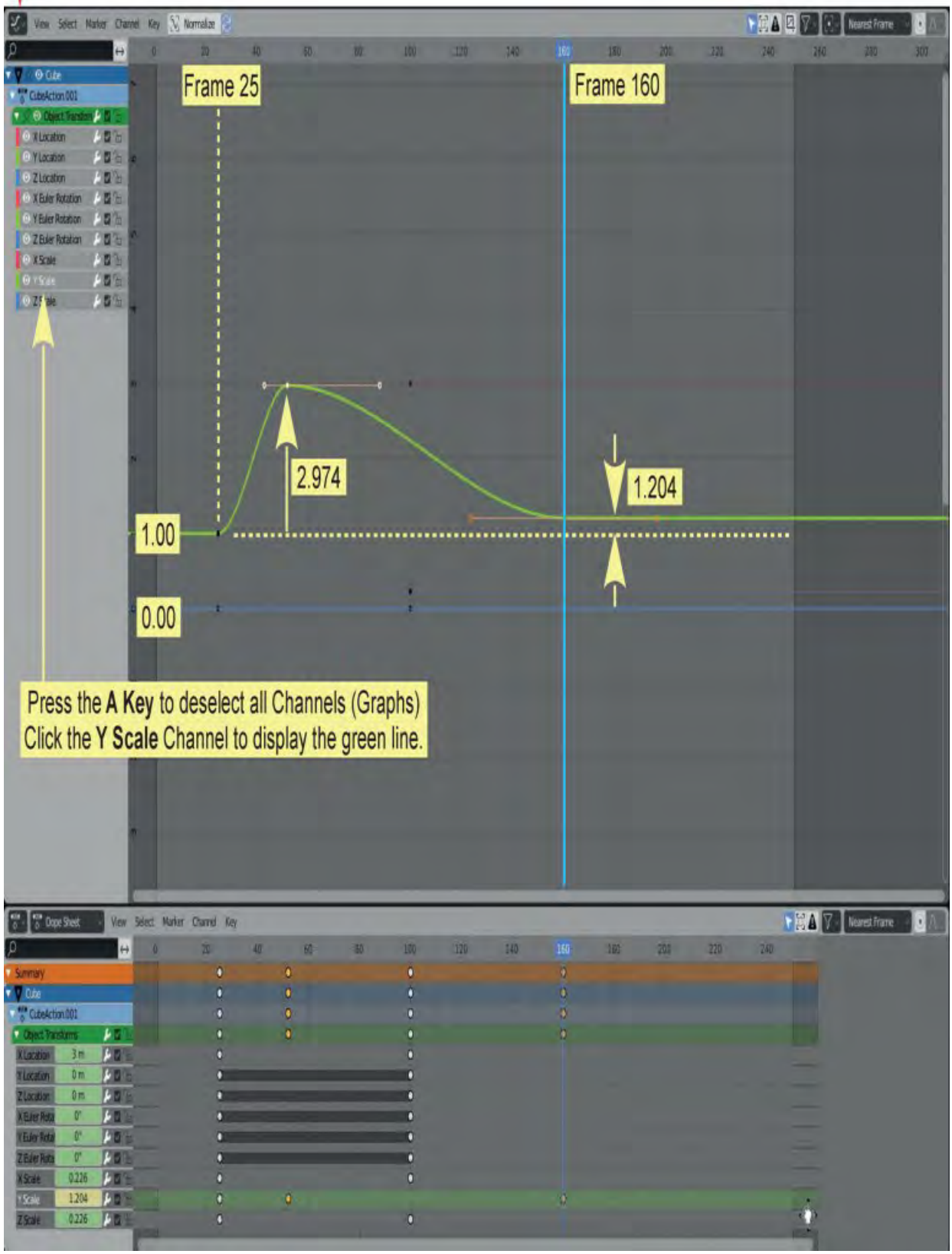


Figure 20.20

The green line in the Graph Editor represents the **Y Scale** Channel in the Dope Sheet Editor which controls the elongation of the Cube Object in the 3D Viewport Editor. Between Frame 1 and Frame 25 in the Animation the Cube is 1.00 X 1.00 X 1.00 (The default Cube). From Frame 25 to Frame 52 the Cube was extended along its Y Axis 2.974 Units then From Frame 52 to Frame 160 the Cube was reduced along its Y Axis to 1.204 Units.

Note: The Graph Line is a **Bezier Curve** (Chapter 10 — 10.2) with Control Handles located corresponding to Keyframes in the Animation Timeline. You may LMB click a Control Handle to select then G Key (Grab), drag the Mouse to reposition the Handle. This reshapes the Graph Line altering the properties of the Object, in this case, affecting the length of the Cube on the Y Axis.

Another example of control using the Graph Editor would be to reverse the direction of travel of an Object in the 3D Viewport Editor. Set up an Animation to have the default Cube Object move from the back of the Scene in the 3D Viewport Editor and travel towards the front of the Scene.

In Figure 20.21 the default Cube moves from back to front in 60 Frames.

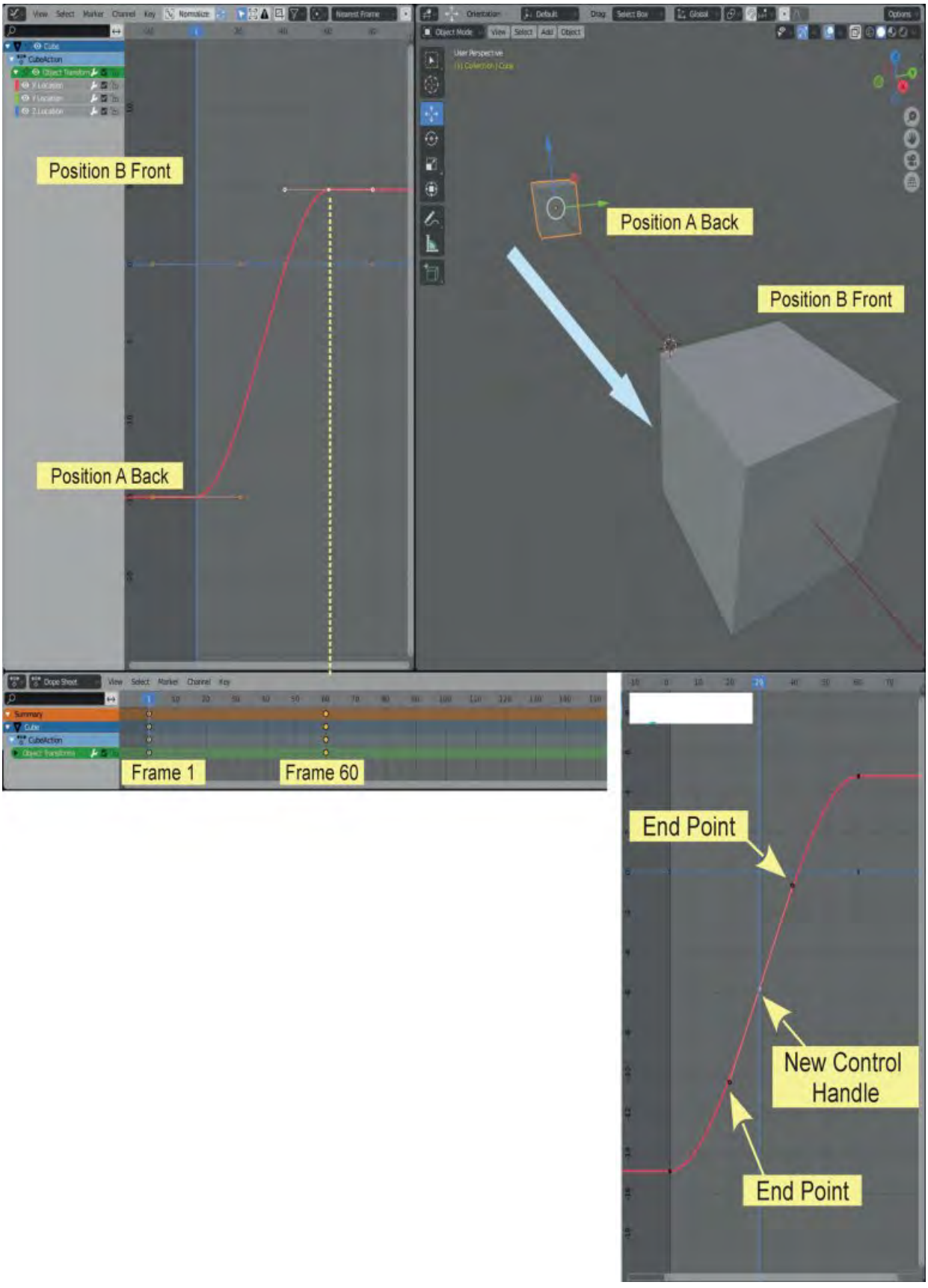

Position B Front
Position A Back
Position A Back
Position B Front
Frame 1
Frame 60
End Point
New Control Handle
End Point

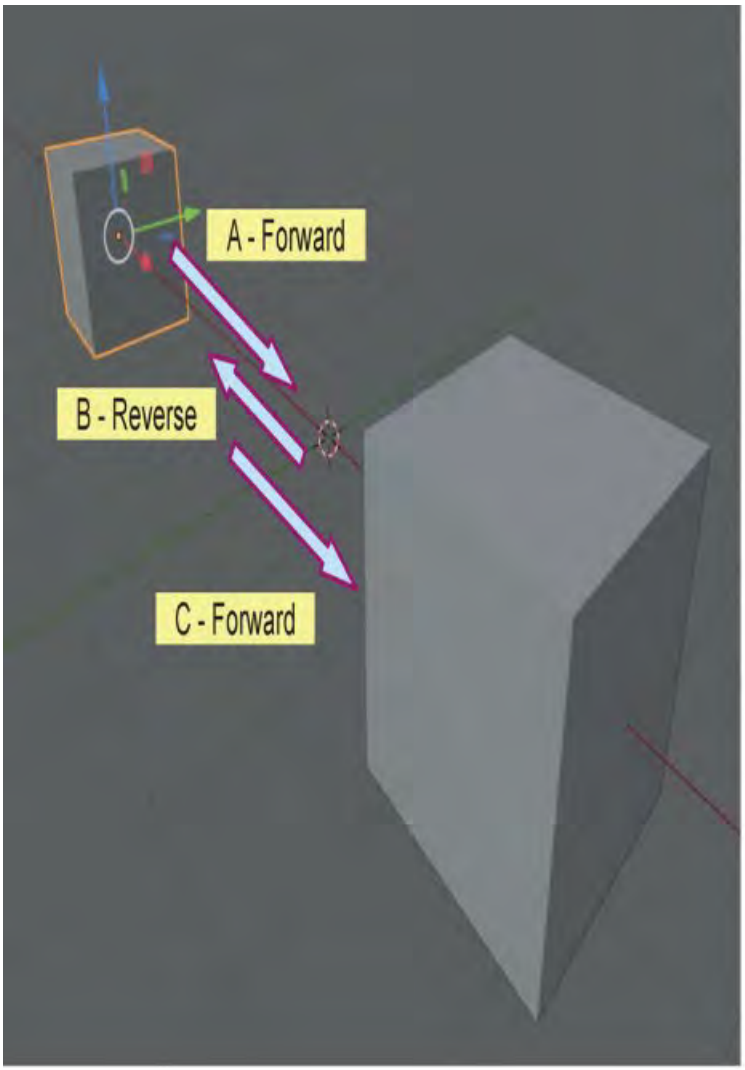
Figure 20.21
Figure 20.22

Inserting Keyframes at Frame 1 and Frame 60 and Translating the default Cube along the X Axis of the Scene produces the Bezier Curve Graph Line shown in the Graph Editor. When the Animation is played the Cube simply moves from the back of the Scene to the front.

With the X Location Channel selected move the Timeline Cursor to a mid point in the Animation and with the Mouse Cursor in the Graph Editor, RMB click and select Insert Keyframes from the menu that displays. Keyframes are inserted and a new Control Handle is placed on the Graph Line (Figure 20.22).

Select the end points of the Control Handle, G Key (Grab) and rotate to shape the Graph Line (Figure 20.23).

With the Graph Line reshaped with a pronounced S-bend the direction of travel of the Cube is reversed during the Animation when the Timeline Cursor intersects the bend.


A - Forward
B - Reverse
C - Forward

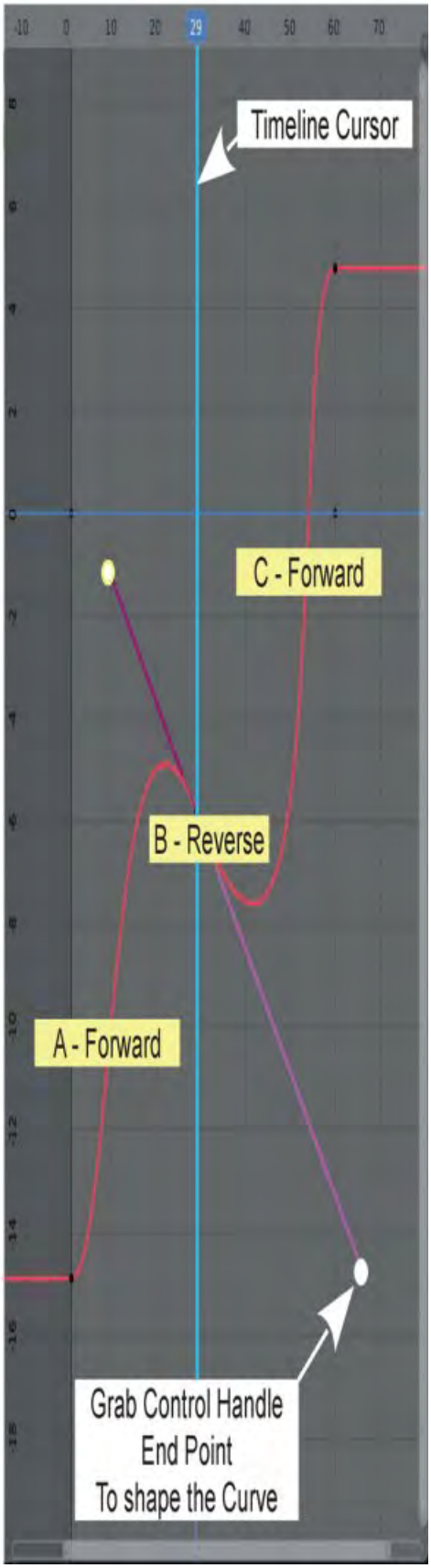

-10
0
10
20
29
40
50
60
70
Timeline Cursor
C - Forward
B - Reverse
A - Forward
Grab Control Handle
End Point
To shape the Curve

Figure 20.23

At this point the Graph Editor has been demonstrated depicting the motion of a Cube Object on the X Axis in the 3D Viewport Editor (the red line in the graph). The red line represents the X Location Channel. When inserting Keyframes in the Animation, selecting Location, Rotation & Scale you set up X, Y and Z Channels for each action (Figure 20.24).

With the default Cube in the 3D Viewport Editor Translated on the X and Y Axis, Rotated about the Z Axis and scaled down the Channels for each can be seen in the Timeline Editor Summary (Figure 20.24).

Figure 20.24

Orange Channels indicate that there was no change in state during the Animation.

The Graph Editor display for the preceding Animation is shown in Figure 20.25. Make note that the Animation Length is the default 250 Frames but the change in state of the Cube occurs between Frame 25 and Frame 100. The Dope Sheet Editor below is aligned with the Graph Editor by adjusting the Editor horizontal Scales and Panning the Views.

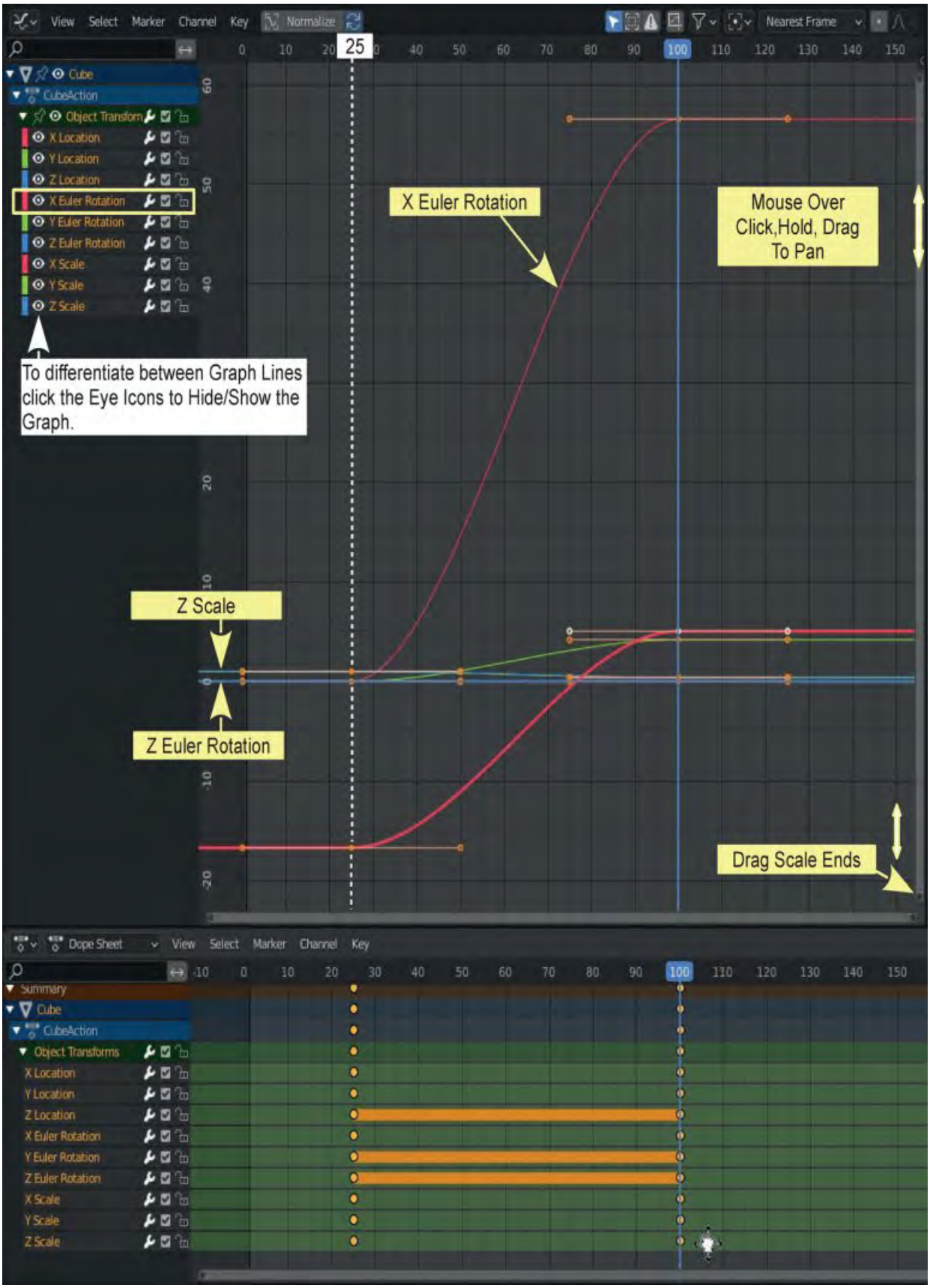

View Select Marker Channel Key Normalize Nearest Frame
Cube
CubeAction
Object Transform
X Location
Y Location
Z Location
X Euler Rotation
Y Euler Rotation
Z Euler Rotation
X Scale
Y Scale
Z Scale
To differentiate between Graph Lines click the Eye Icons to Hide/Show the Graph.
X Euler Rotation
Mouse Over Click,Hold, Drag To Pan
Z Scale
Z Euler Rotation
Drag Scale Ends
Dope Sheet View Select Marker Channel Key
Summary
Cube
CubeAction
Object Transforms
X Location
Y Location
Z Location
X Euler Rotation
Y Euler Rotation
Z Euler Rotation
X Scale
Y Scale
Z Scale

Figure 20.25

Examine the Graph Editor in more detail (Figure 20.26 over).

Note: In Figure 20.26 the Animation occurs between Frame 25 and Frame 75.

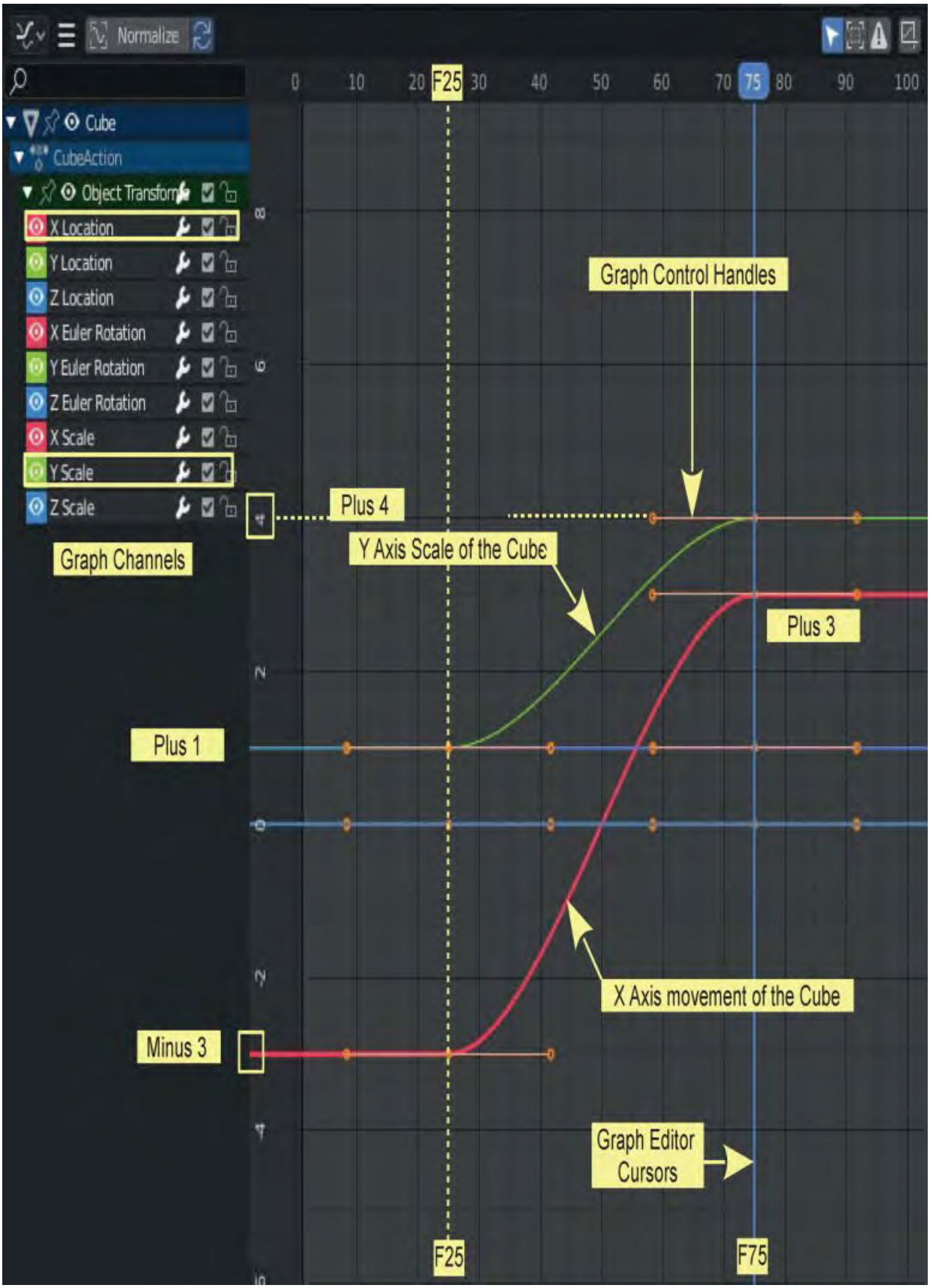

Normalize
0
10
20
F25
30
40
50
60
70
75
80
90
100
Cube
CubeAction
Object Transform
X Location
Y Location
Z Location
X Euler Rotation
Y Euler Rotation
Z Euler Rotation
X Scale
Y Scale
Z Scale
Graph Channels
Graph Control Handles
Plus 4
Y Axis Scale of the Cube
Plus 3
Plus 1
X Axis movement of the Cube
Minus 3
Graph Editor Cursors
F25
F75

Figure 20.26

In the Graph Editor the red line represents the movement of the Cube Object on the X Axis of the 3D World. The movements initial position of minus three units changes to plus three units between Frame 25 and 75 (the position of the Keyframes). The green line represents the change in Scale of the Cube on the Y Axis (default Cube Scale = 1. Scale value entered = 4). The two horizontal blue lines represent the Z Axis Location and Scale which, being horizontal, means no change. You see Frame numbers in the scale at the top of the Editor and Blender units in the vertical scale. Note: The vertical alignment of graph lines with the scale is approximate.

The Graph Channels in the panel at the LH side of the Editor list the actions that have been entered in the animation by inserting Keyframes. When entering Keyframes, type **Location, Rotation & Scale** was selected, therefore, the Graph Channels list Location, Rotation and Scale actions. In the Cube animation only the X Location and Y Scale Channels are of significance. The remaining Channels are shown since Location, Rotation & Scale was the Keyframe Type selected.

20.15 Editor and Commands

Graph Editor Cursors is the **Vertical blue line** — LMB click on the blue button, hold and drag in the Frame Header to position.

Scaling the Graph Editor may be done by clicking on the **Scale Bar,** holding and dragging the Mouse. Click and hold a dot at either end of a scale to shrink or extend the scale.

Scaling the Frame Bar occurs when the Scale Bar is manipulated.

The Dope Sheet Channels, Cube, Cube Action and Object Transforms may be expanded or collapsed by clicking the white triangle preceding each name. (Figure 18.27).

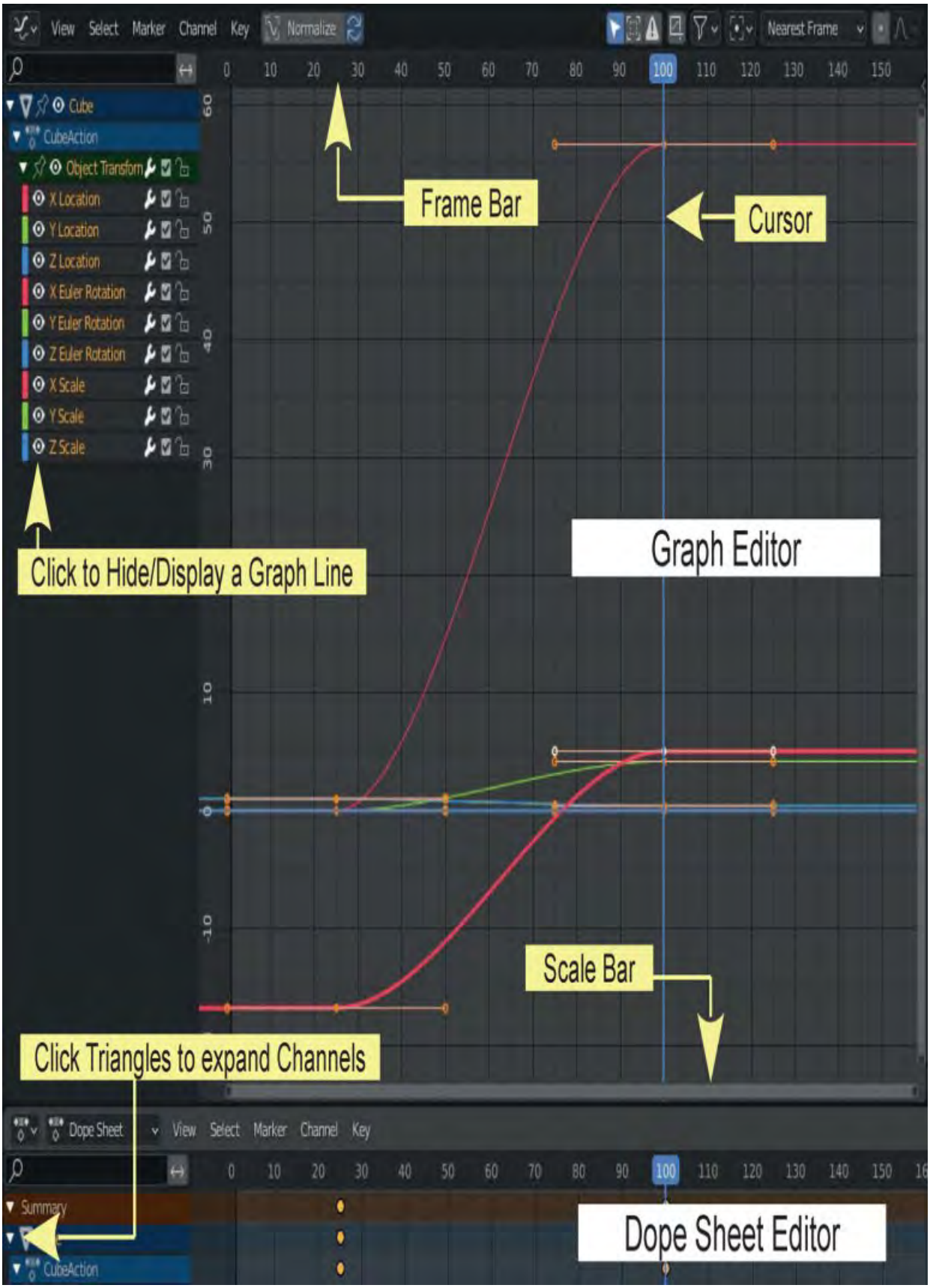

View Select Marker Channel Key Normalize Nearest Frame
Cube
CubeAction
Object Transform
X Location
Y Location
Z Location
X Euler Rotation
Y Euler Rotation
Z Euler Rotation
X Scale
Y Scale
Z Scale
Frame Bar
Cursor
Graph Editor
Click to Hide/Display a Graph Line
Scale Bar
Click Triangles to expand Channels
Dope Sheet View Select Marker Channel Key
Summary
CubeAction
Dope Sheet Editor

Figure 20.27

Hiding Graphs: Clicking the eye icons preceding each Graph Channel toggles hide and display of the graph lines in the Editor.

Collapsing Channels in the Dope Sheet and hiding graph lines in the Editor can be very useful in a complicated animation. In the demonstration only the X Axis Translation and Y Axis Scale have been animated. There are many more features of the Cube alone which may be animated, therefore, you can imagine the Graph Editor could become congested with information.

Positioning the Graphs relative to the Timeline in the Graph Editor may be accomplished by selecting Control Handles on a Graph Line, pressing the G Key and dragging the Mouse. This action moves the selection, moving the Graph Line. The movement is relative to the Timeline. Graph Lines that are not selected remain stationary, therefore, the movement of the selected line alters the action of the Object in the 3D View Editor relative to other actions.

20.16 Selecting the Curve in the Graph Editor

As previously explained, the X Location Curve in the Graph Editor is a Bezier Curve representing the X Axis movement of the Cube, therefore, the Control Handles can be selected and manipulated. The following commands execute selection procedures when the Mouse Cursor is in the Graph Editor panel:

Press **Alt + A Key** deselects the Control Handles. Press the **A Key** reselects the Control Handles.

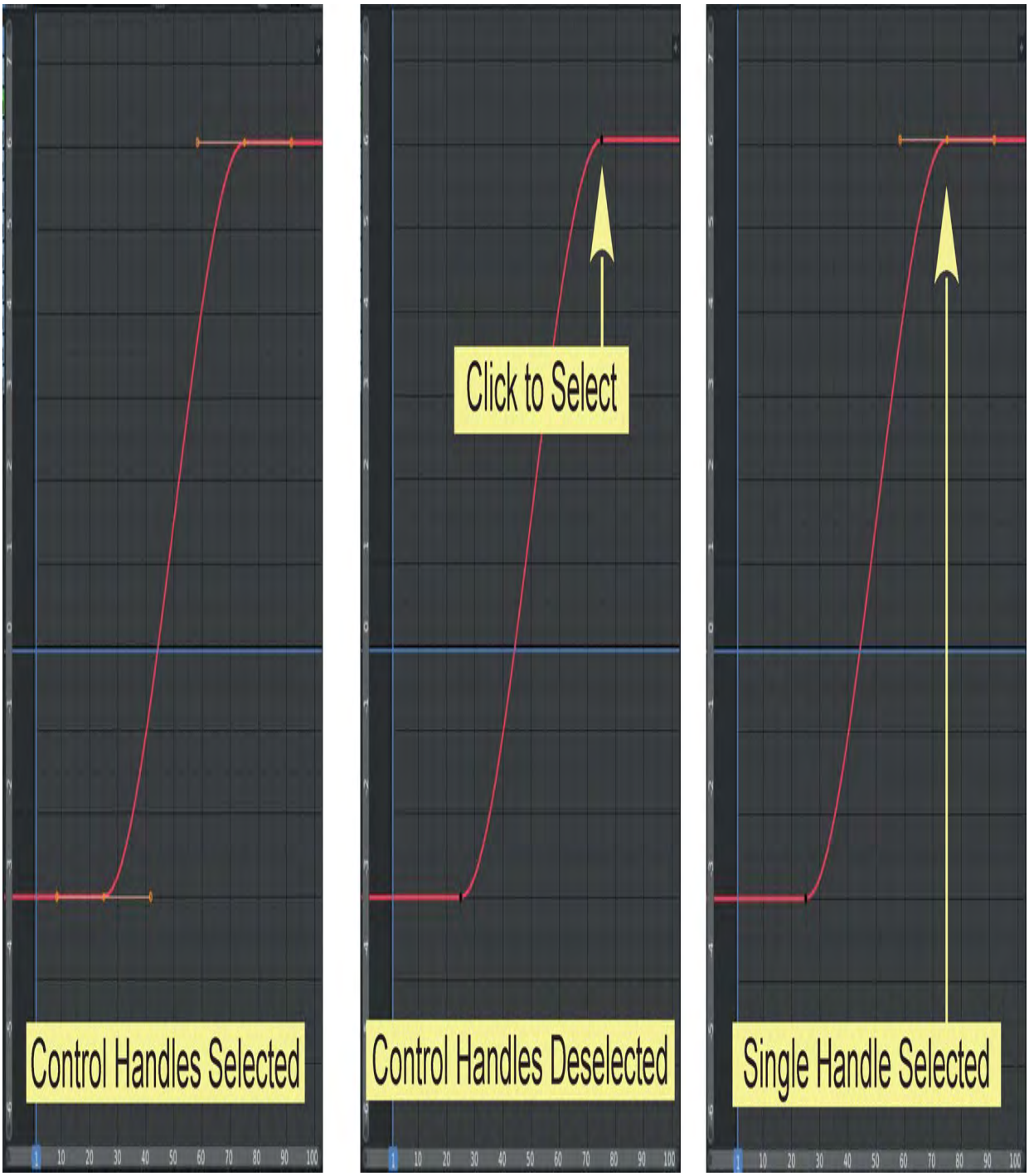


Figure 20.28

With Control Handles deselected (click LMB in an empty space), click LMB on a Control Handle (black dot) to select. With a Control Handle

selected it may be Translated, Rotated and Scaled by: **G** Key (Grab), move the Mouse, LMB click to locate.

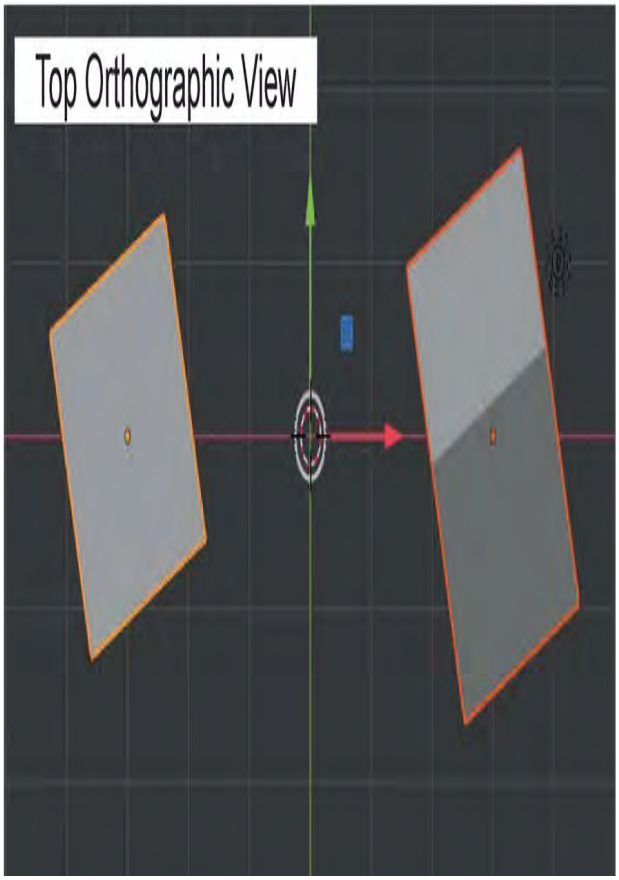
R Key (Rotate) move the Mouse, LMB click to set.

S Key (Scale) move the Mouse, LMB click to set.

Each of the above actions reshapes the Curve affecting the X Axis movement of the Cube in the animation. With all Control handles selected or deselected, pressing the Tab Key leaves the Curve displayed as a broken line which is uneditable.

20.17 Rotation – Euler Rotation.

You will have observed that Rotation values in Blender are labelled **Euler Rotation**. Simply put, **Euler**, is a method of expressing Rotation in a **Three Dimensional Environment**.



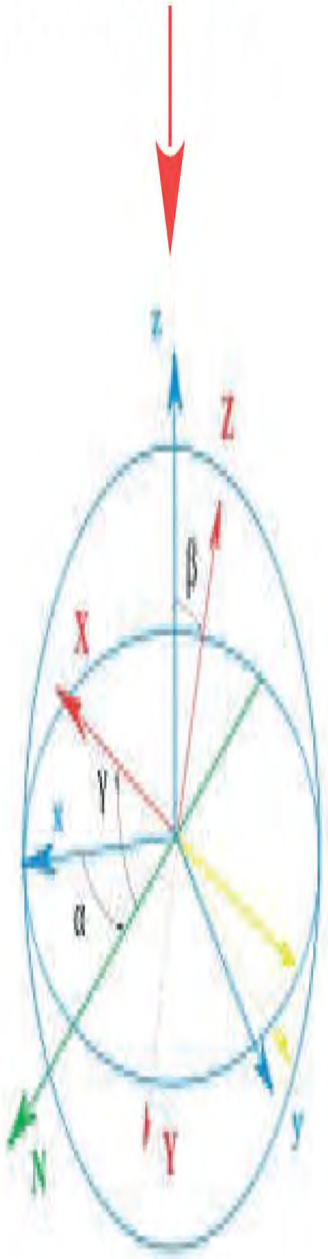


Figure 20.30

In Figure 20.29 the Cube on the left is Rotated 20 on the Z Axis of the Scene. The Cube on the right is also Rotated 20 on the Z Axis but prior to this Rotation it has been Rotated 37.5 on the X Axis of the Scene.

G Key (Grab), move the Mouse, LMB click to locate.
R Key (Rotate) move the Mouse, LMB click to set.
S Key (Scale) move the Mouse, LMB click to set.

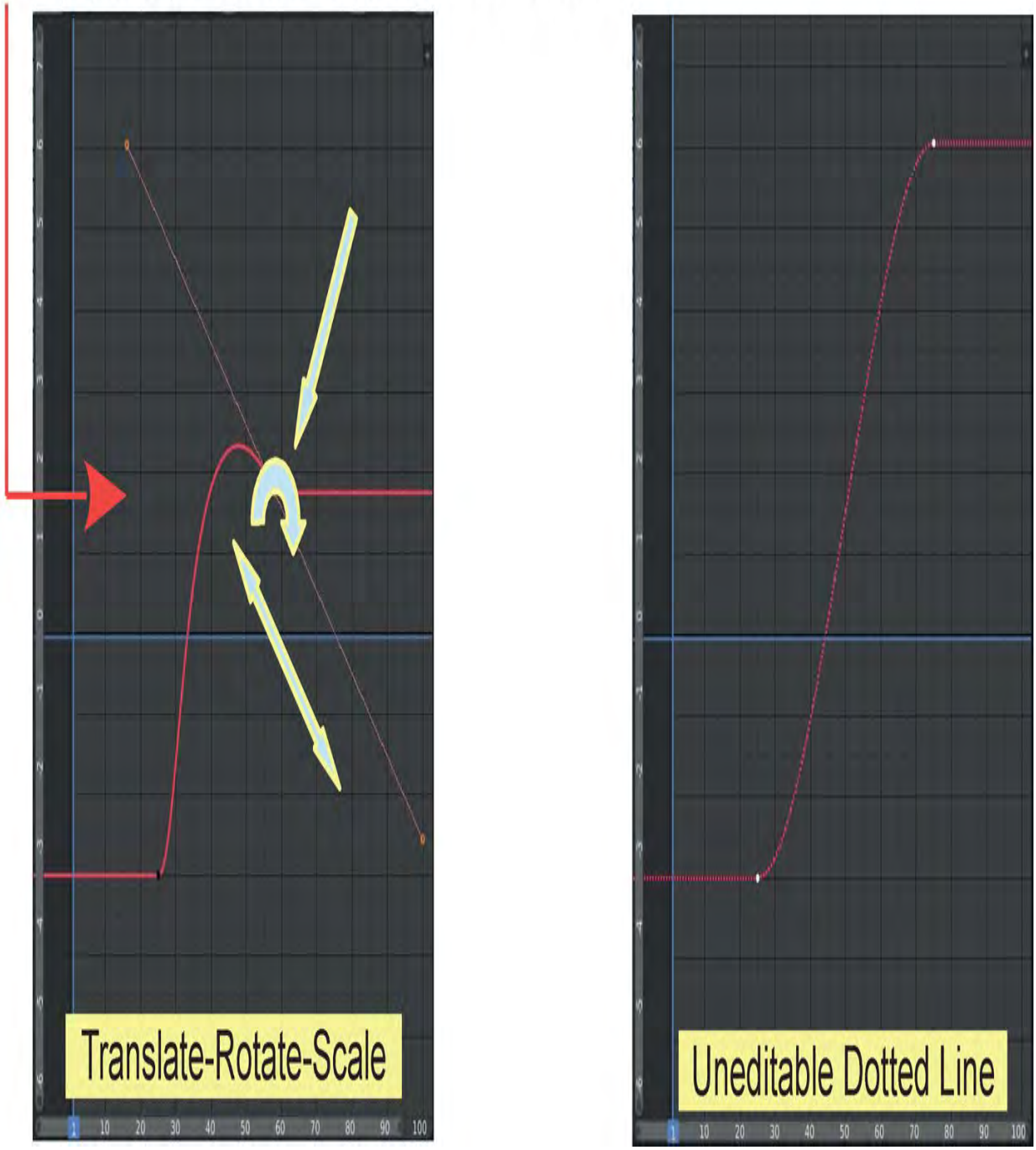


Figure 20.29

Rotation can be made about the **Local Axis** of the Object by selecting Local Axis in the **Transformation Orientation** options in the 3D Viewport Editor Header.

Rotation can be made about the **Local Axis** of the Object by selecting Local Axis in the **Transformation Orientation** options in the 3D Viewport Editor Header.

Euler Rotation, therefore, defines the combination of Global Rotation and Local Rotation.

Figure 20.31

Euler Rotation, therefore, defines the combination of Global Rotation and Local Rotation.

20.18 The Graph Editor Cursor

Before proceeding, how to control the Graph Editor Cursor should be explained.

With the default Cube Object in the 3D Viewport Editor, before any animation has taken place the Graph Editor displays as shown in Figure 20.32. There is a vertical blue line Cursor at Frame 1 in the Timeline and a horizontal blue line Cursor at position 0 (zero) in the vertical Scale.

Default Graph Editor Display Timeline - Frames

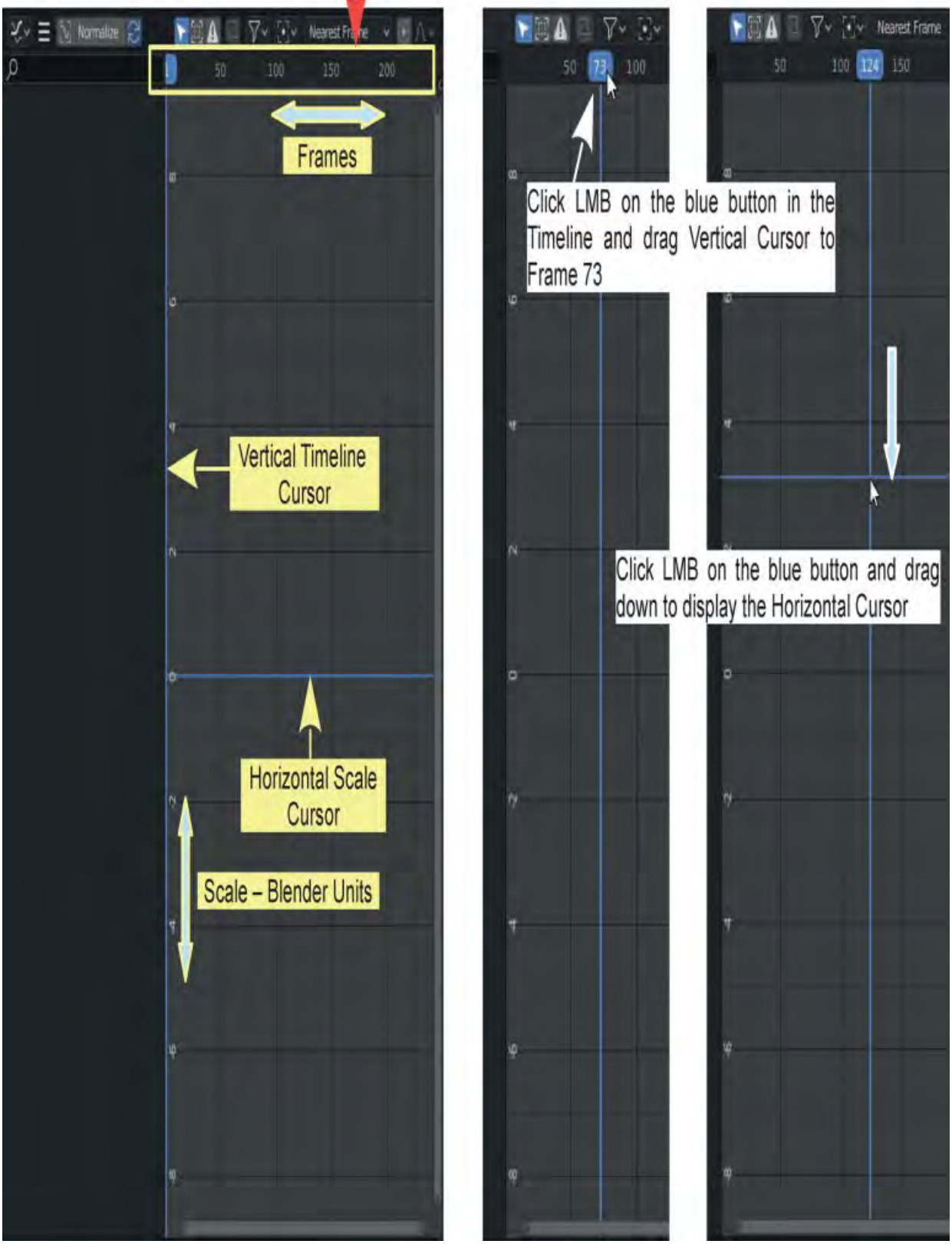


Figure 20.32

When you click LMB on a Frame in the horizontal Timeline or click, hold and drag the blue Cursor button in the Timeline, moving the vertical Cursor, the horizontal blue line Cursor disappears from view. Moving the vertical Cursor back to Frame 1 does not reinstate the horizontal Cursor.

To display the horizontal Cursor, click on the blue button in the Timeline, hold and drag down. By holding LMB and dragging both vertical and horizontal Cursors are positioned in the Graph Editor.

The significance of positioning the Cursors will follow.

20.19 Editing the Curve in the Graph Editor

To edit the X Axis movement of the Cube, select the top Control Handle, press the G Key and move the handle down three units and left towards the Cursor with Frame 50. Click **LMB** to set in position (Figure 20.33).

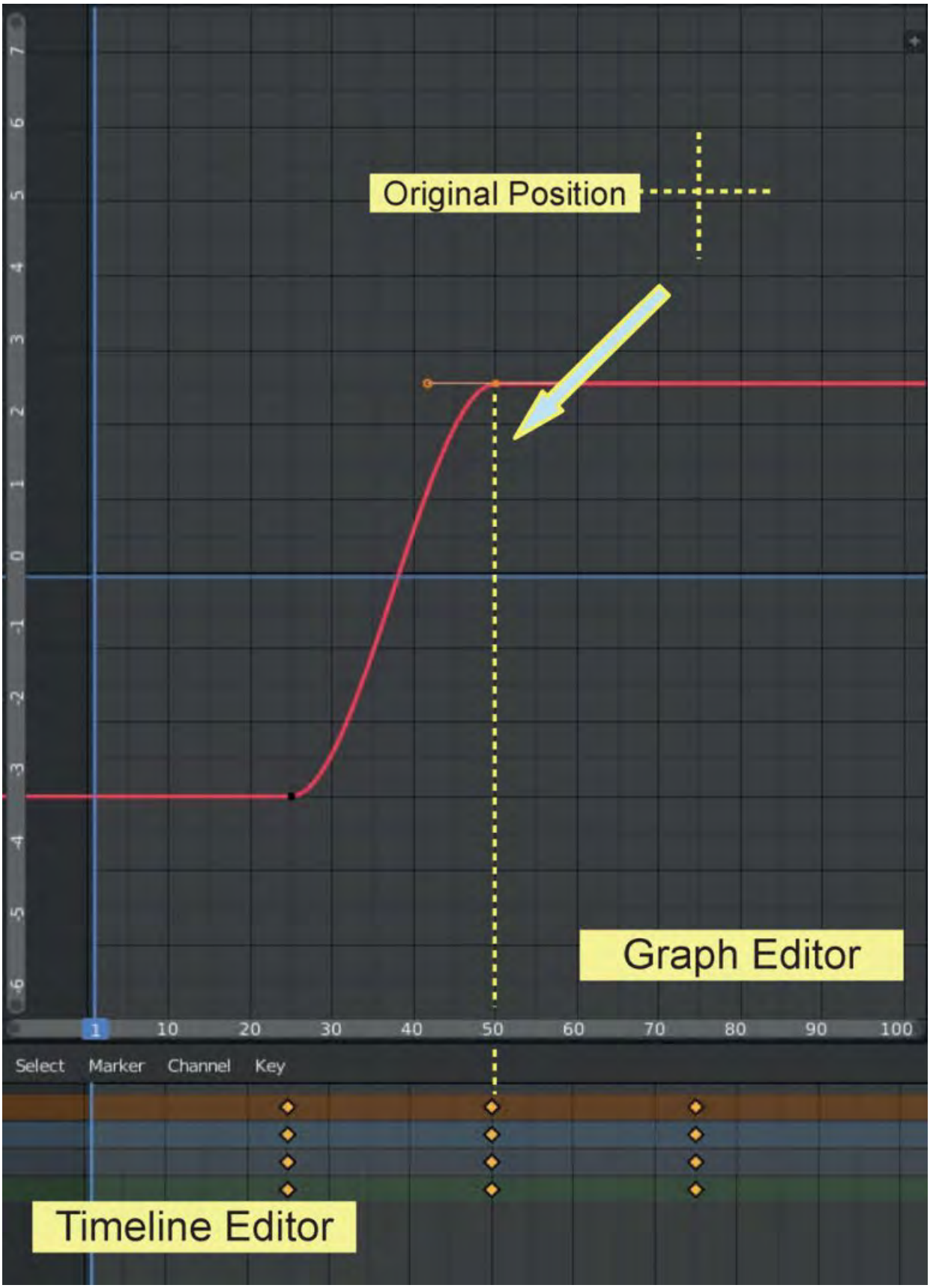

Original Position
Graph Editor
1
10
20
30
40
50
60
70
80
90
100
Select
Marker
Channel
Key
Timeline Editor

Figure 20.33

Note: When the Control Handle is moved you will see a set of Keyframes move to the Frame where you position the handle (Frame 50). This set of Keyframes is for the X Axis movement of the Cube. The Keyframes remaining at Frame 75 are for the Y Axis scale of the Cube.

Playing the animation at this point will see the Cube start to move on the X Axis at Frame 25 then stop at Frame 50. The Cube will scale on the Y Axis between Frame 25 and frame 75. Before frame 25 and after frame 75 no change in state occurs.

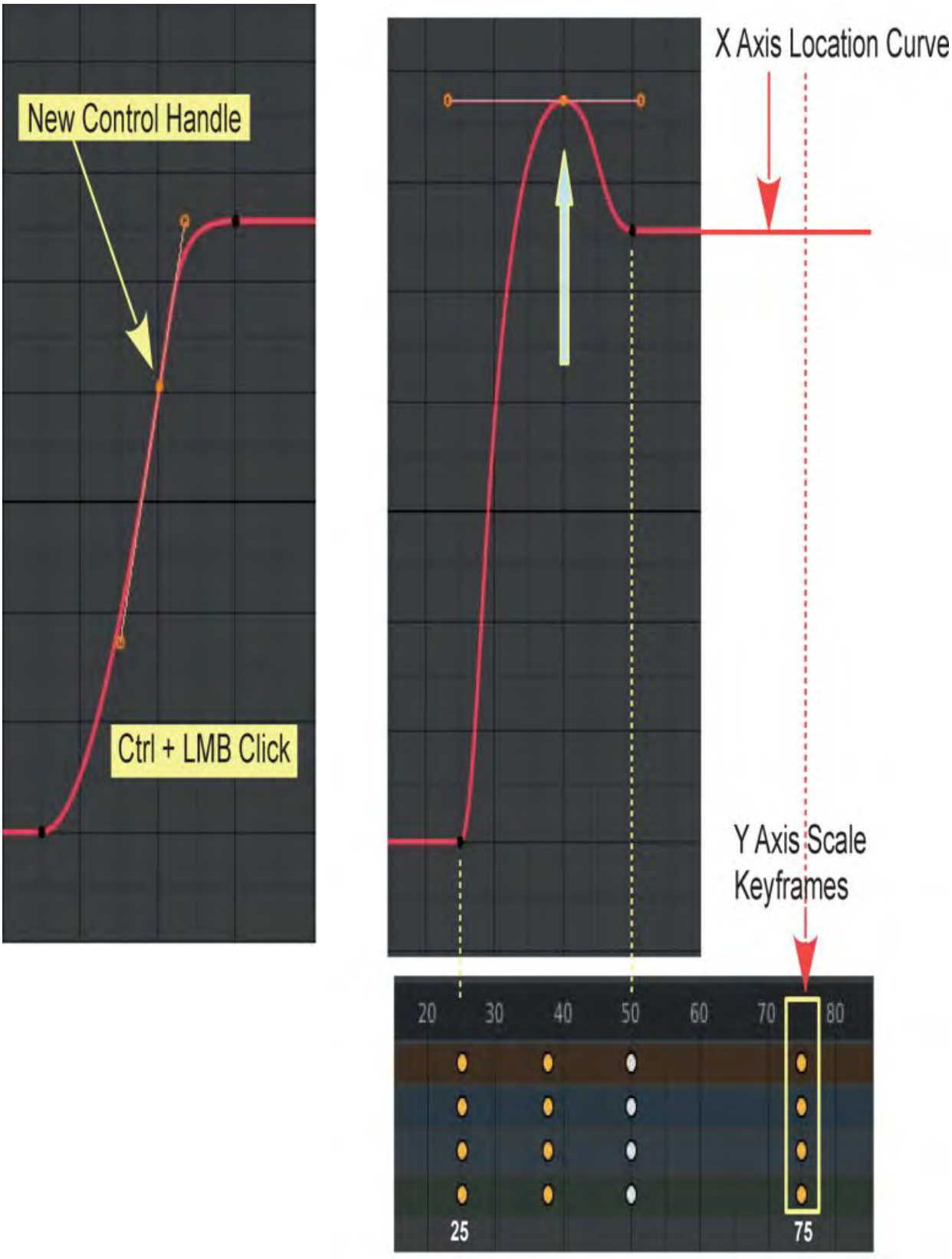
By scaling and rotating the top Control Handle you further edit the Curve (Figure 20.34.

With the handle shaping the Curve as shown in Figure 20.34, when the animation is played, the Cube moves forward on the X Axis between Frame 25 and Frame 44 then reverses direction until frame 50. **Note:** With this method no additional Keyframes are added to the Timeline.

F44
F50

Figure 20.34

Alternatively, instead of rotating the top Control handle, press **Ctrl** and **RMB Click on the Curve** to add a new Control Handle (Figure 20.35). New Keyframes are inserted in the Timeline. With the handle selected press G Key and Translate it up and Scale approximating the Curve previously created.



Keyframes in the Dope Sheet Timeline

Figure 20.35

Using this alternative method gives more control in editing since it provides an additional Control Handle and Keyframes.

Note: The Keyframes in the Timeline.

Yellow Keyframes are selected.

White Keyframes are deselected.

Keyframes in the Dope Sheet Timeline

20.20 Editing the Curve in the Dope Sheet Timeline

Curves and, therefore, the animation may be edited, by repositioning Keyframes in the Timeline of the Dope Sheet Editor. The first operation is to select individual Keyframes.

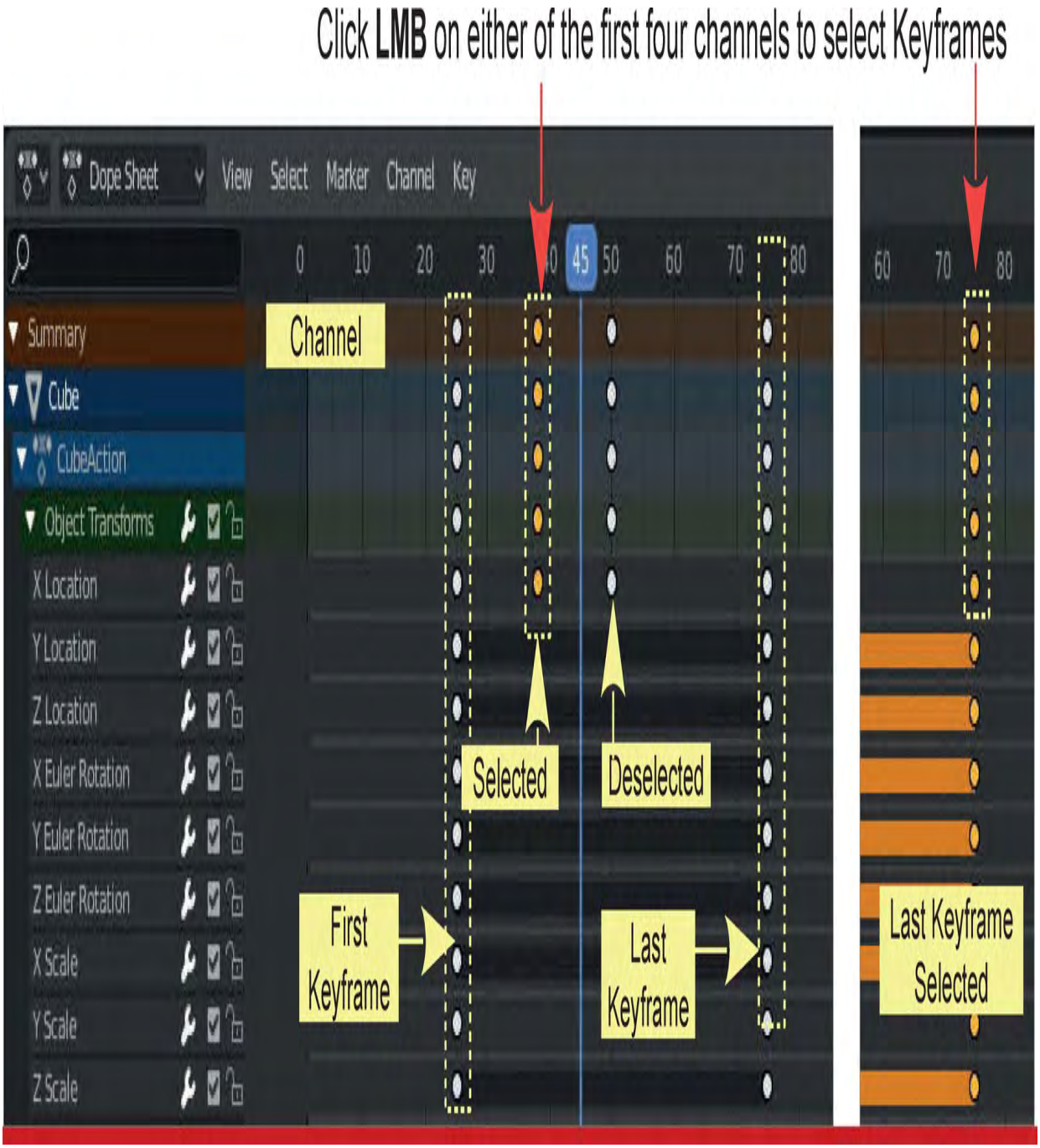


Figure 20.36

Selecting Keyframes in the Dope Sheet Editor follows the basic rules for all Editors. **LMB** Click to select. Press **Alt + A Key** to deselect. Press **A Key** to select all.

With a Keyframe selected press **Delete** or the **X Key** to delete the Keyframe.

With a Keyframe selected press the **G Key,** hold **LMB** and drag to reposition the Keyframe.

With the **Object Transforms** expanded, click **LMB** on a Channel to select it (Z Location). Click **LMB** on a Keyframe in the Channel, Press **G Key,** hold **LMB** and reposition.

First Z Location Keyframe Repositioned
Dope Sheet
View
Select
Marker
Channel
Key
1
10
20
30
40
50
60
70
80
Summary
Cube
CubeAction
Object Transforms
X Location
Y Location
Z Location
X Euler Rotation
Y Euler Rotation
Z Euler Rotation
X Scale
Y Scale
Z Scale
Mouse Cursor
In Grab Mode
F33

Figure 20.37

Note: Repositioning the Z Location Keyframe as shown has no effect on the Cube in the 3D View Editor. When inserting the LocRotScale Keyframe no change was made to the Z Location in the 3D View Editor, therefore, the Z Location Graph is a horizontal straight line.

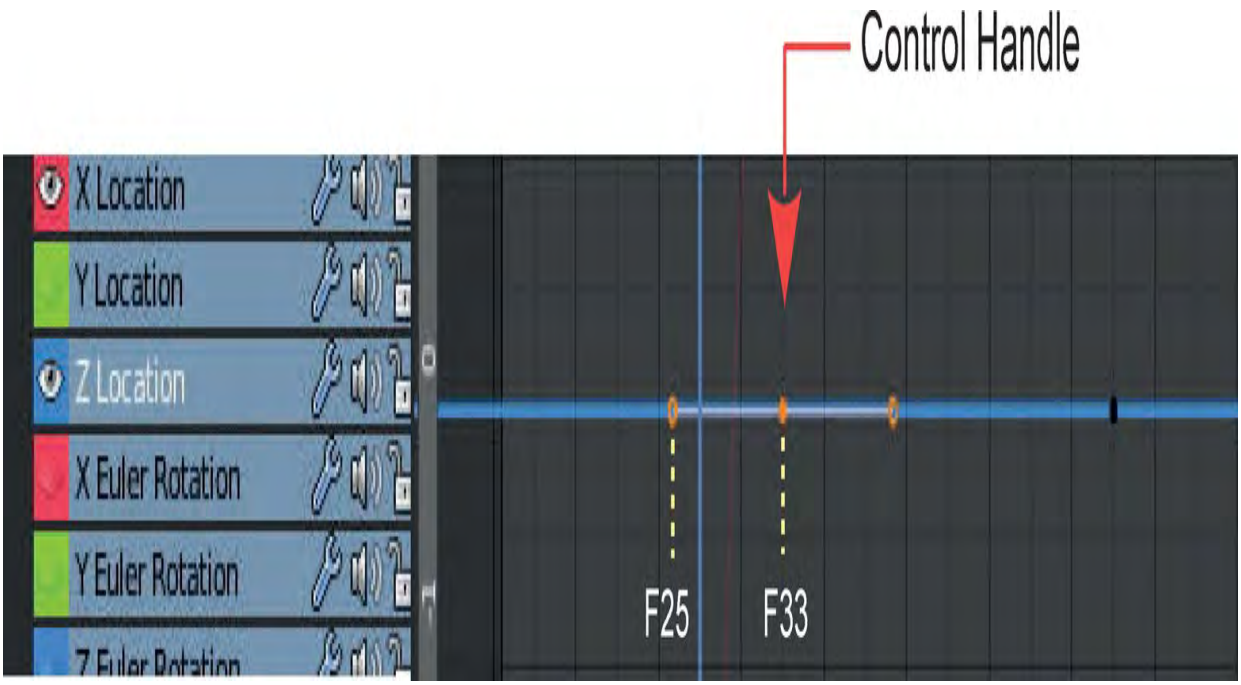


Figure 20.38

Repositioning the Keyframe in the Dope Sheet has repositioned the Control handle in the Graph Editor which, if there were a change in elevation on the Z Axis in the scene for the Cube, it would commence at this point.

To understand the correlation between the Graph Editor and the Dope Sheet align the two Editors one above the other (Figure 20.39 over).

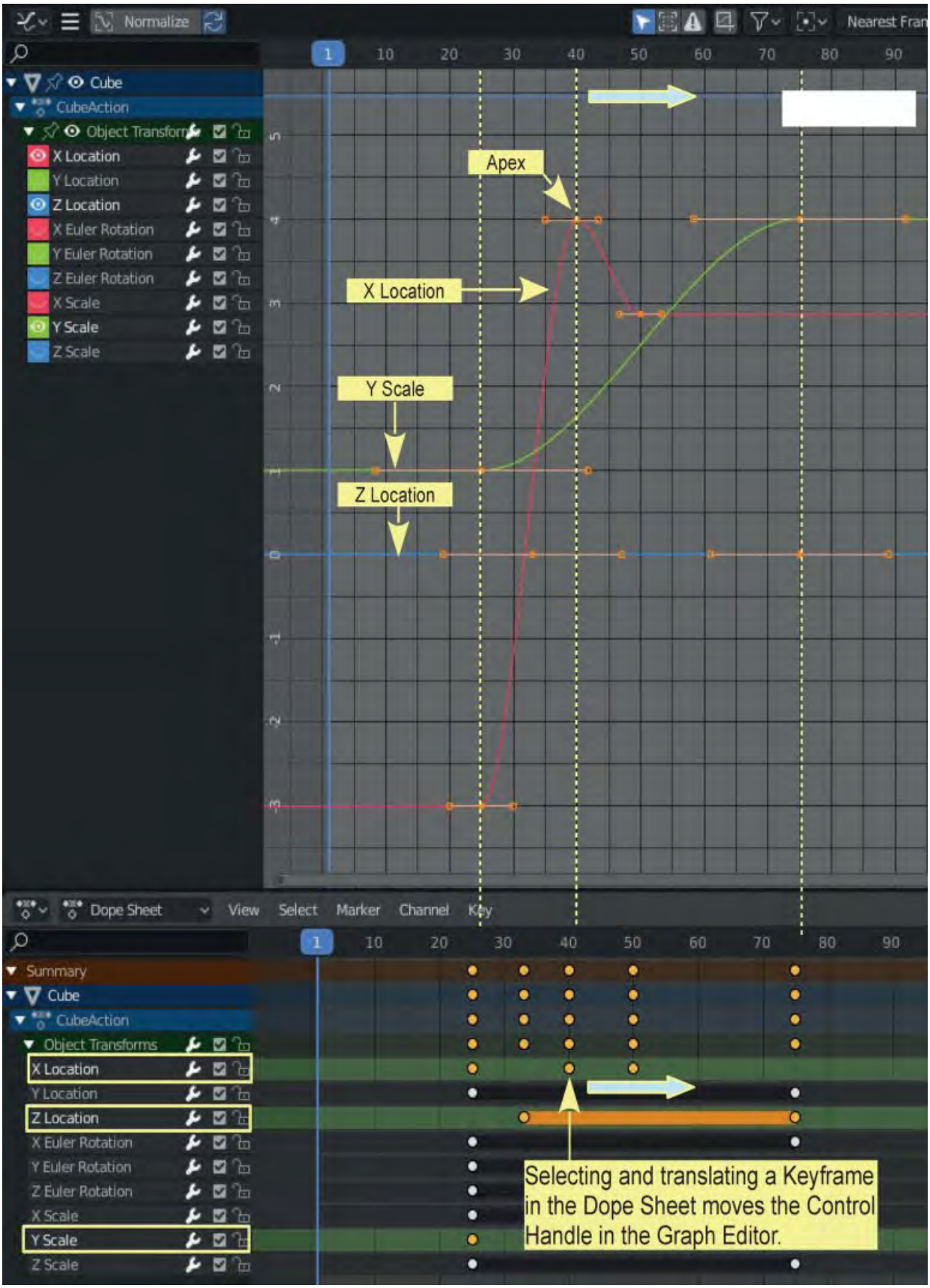

Normalize
Nearest Fran
Cube
CubeAction
Object Transforms
X Location
Y Location
Z Location
X Euler Rotation
Y Euler Rotation
Z Euler Rotation
X Scale
Y Scale
Z Scale
Apex
X Location
Y Scale
Z Location
Dope Sheet
View
Select
Marker
Channel
Key
Summary
Cube
CubeAction
Object Transforms
X Location
Y Location
Z Location
X Euler Rotation
Y Euler Rotation
Z Euler Rotation
X Scale
Y Scale
Z Scale
Selecting and translating a Keyframe in the Dope Sheet moves the Control Handle in the Graph Editor.

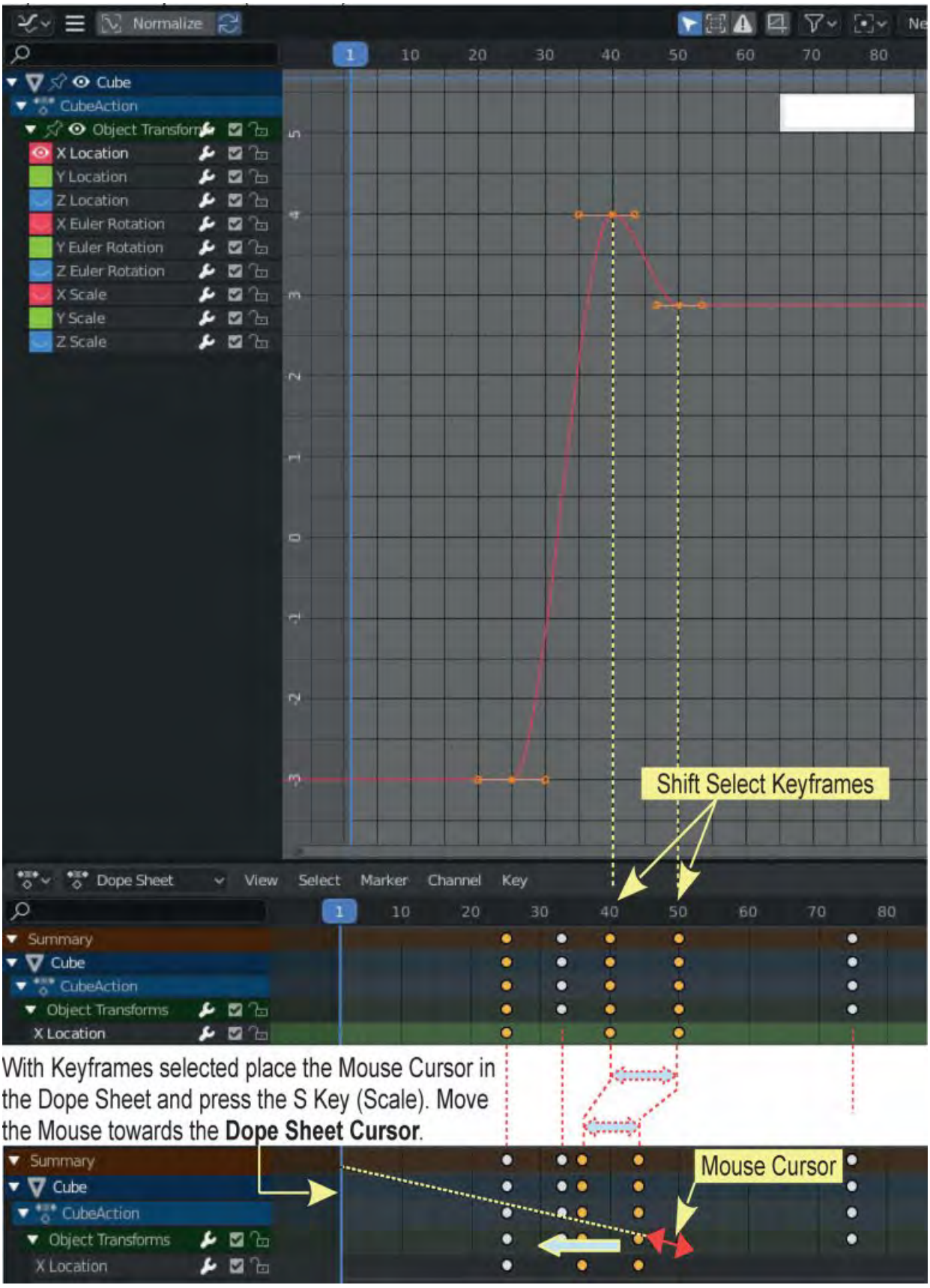
Figure 20.39

Translating a Keyframe in the Dope Sheet moves the Control Handle in the Graph Editor. Where the Keyframe/Control Handle is positioned determines where the action in the animation occurs. In Figure 20.39 the apex of the X Location Curve is the point where the forward movement of the Cube in the 3D Viewport Editor is reversed.

20.21 Scaling in the Dope Sheet

Besides moving Keyframes adjustment to when an action takes place in the animation may also be adjusted by Scaling on the Timeline in the Dope Sheet.

Note: Scaling is relative to the Dope Sheet Editor Cursor. To demonstrate consider the movement of the Cube on the X Axis between the apex of the Curve in the Graph Editor (Frame 40) and the final position (Frame 50).



With Keyframes selected place the Mouse Cursor in the Dope Sheet and press the S Key (Scale). Move the Mouse towards the **Dope Sheet Cursor**.

Figure 20.40
Figure 20.41

The selected Keyframes are Scaled relative to the Dope Sheet Cursor (blue line). You will see the shape of the Curve in the Graph Editor modified accordingly.

Repeat the procedure, this time, positioning the Dope Sheet Cursor between the selected Keyframes. The Scaling is relative to the Cursor.

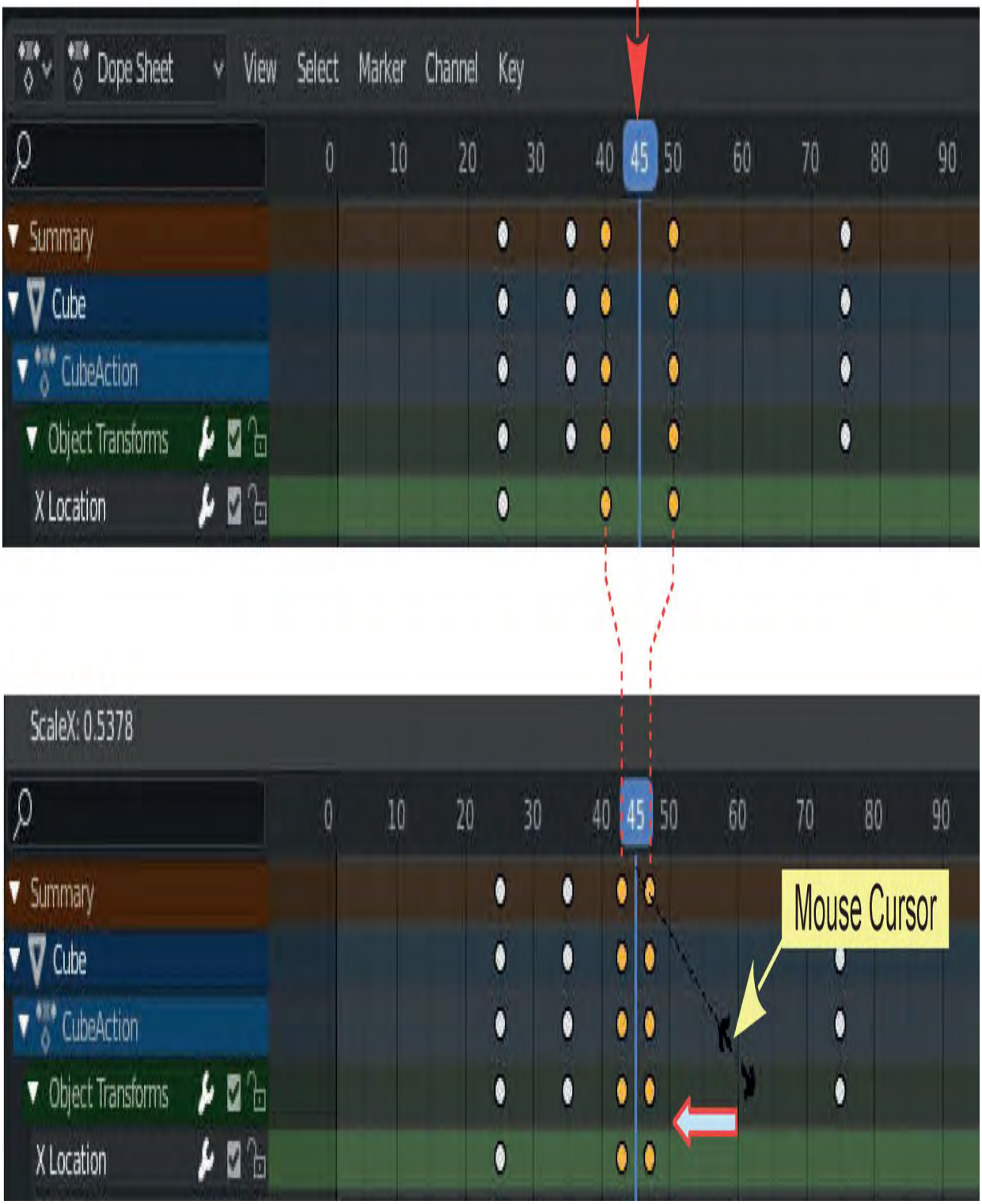


Figure 20.42

Moving Keyframes and manipulating Control Handles edits the animation Curve. The Curve has been a Bezier Type but there are alternative types to be considered which means alternative types of Interpolation between Keyframes resulting in different types of motion for the selected Object.

20.22 Other Types of Curves

By default, Blender displays **Bezier Type Curves** in the Graph Editor which means that **Bezier Type Interpolation** is used between **Keyframes.**

When considering Bezier Type Interpolation (the Curve shape) the curves at either end of the graph line represent the acceleration of the Object.

Other Types of Interpolation (Curves) are accessed in the Graph Editor or the Dope Sheet Editor. In the Dope Sheet Header click **Key** and select **Interpolation Mode** or with the Mouse Cursor in either Editor press the T Key. Either method opens the Set Keyframe Interpolation menu.

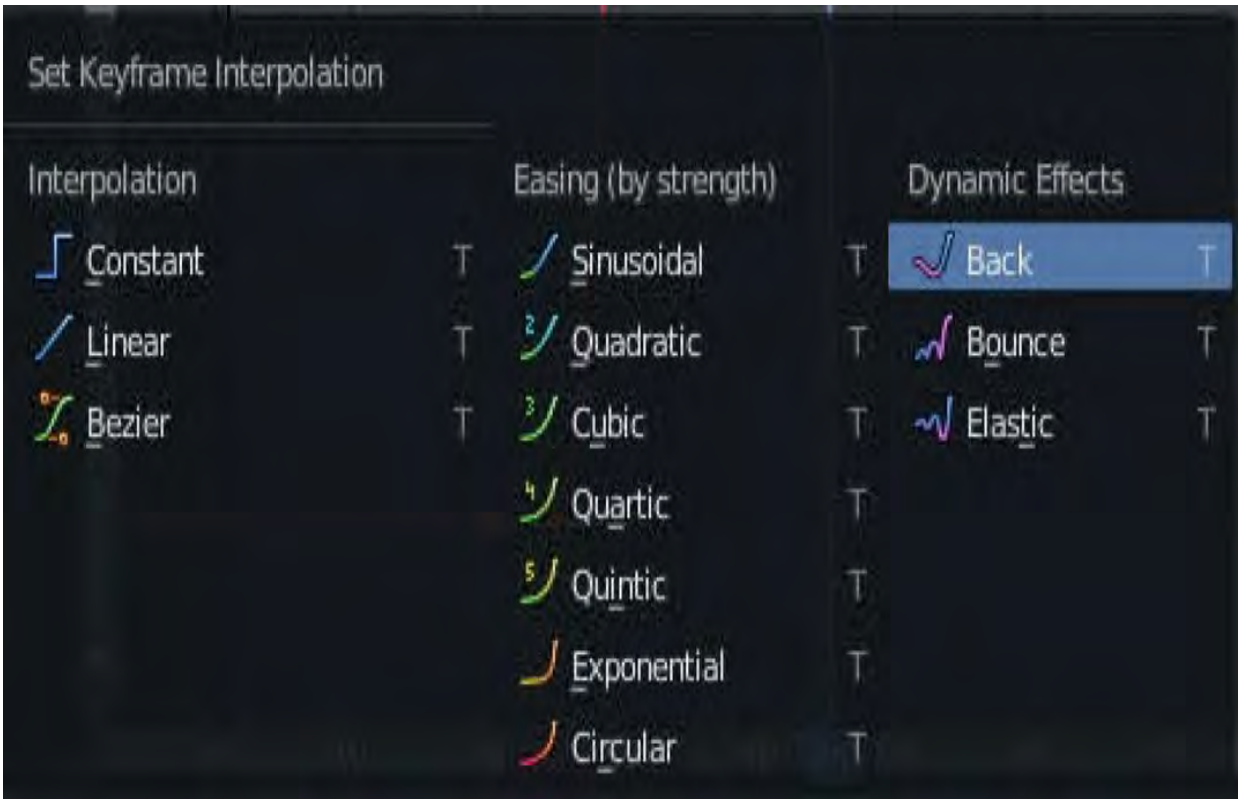


Figure 20.43

At this point only be concerned with the Constant, Linear and Bezier options.

With a Curve selected in the Graph Editor (the default is Type Bezier) select **Constant** or **Linear.**

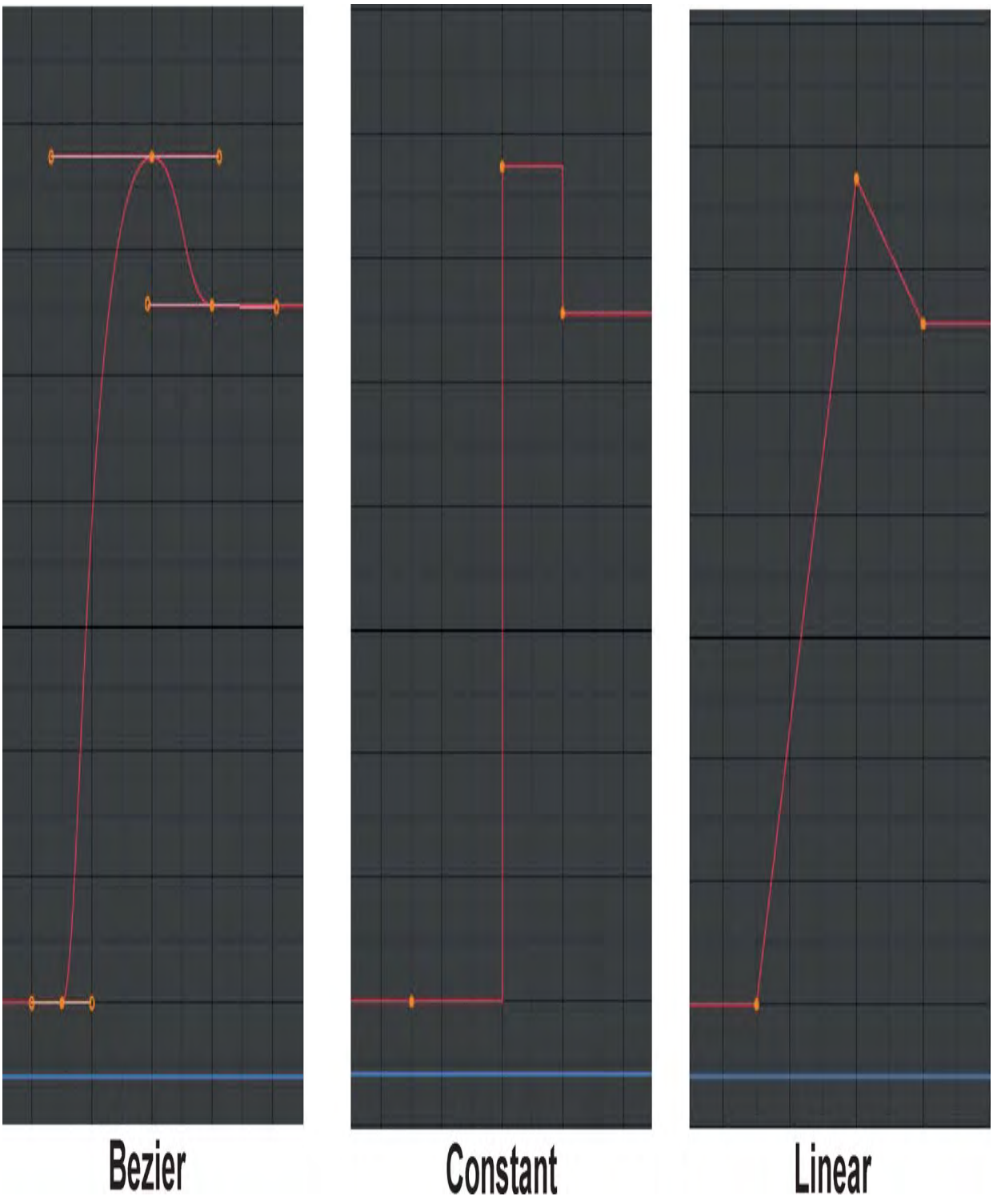


Figure 20.44

Each Type of Interpolation (Curve) produces a different motion in the animation. **Constant Interpolation** results in a dramatic quick change from

one state to the other at a given Frame while **Linear Interpolation** produces a change following a straight line graph between points. The choice of these types of graphs and motions depends on how you want your actor to behave in the animation. Both of the alternatives to Bezier give the option to grab and move points and to add additional points on the graph, but Bezier is by far the most flexible of the three.

Extrapolatio

Blender interpolates to add frames between the Keyframes according to which of the previous Curve options were selected. Blender can also figure out what to do with the frames of the animation before the first Keyframe and after the last Keyframe, which is called **Extrapolation.**

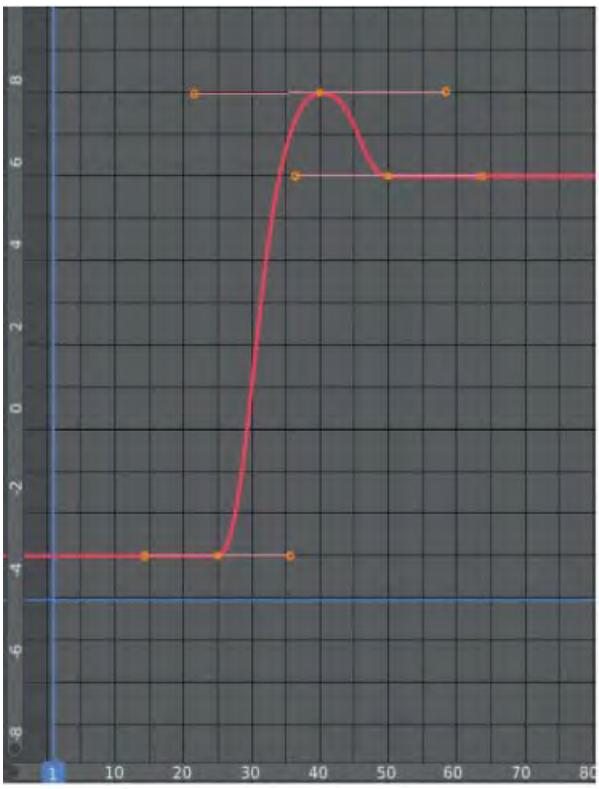
With the Mouse Cursor in the Graph Editor press **Shift + E Key** to display the **Set Keyframe Extrapolation** menu.

Figure 20.45

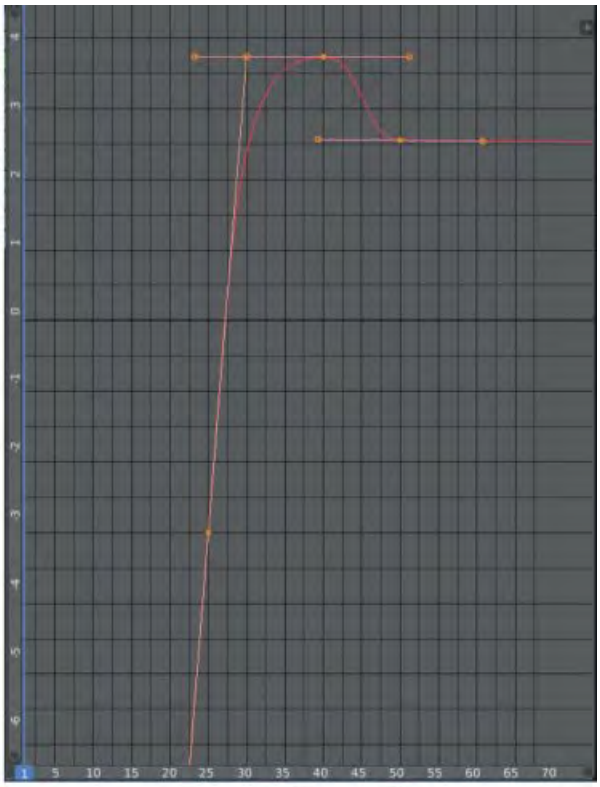
Constant Extrapolation: Blender has inserted frames that comply with a Bezier curve. On either side of the Keyframes, you can see horizontal lines that indicate there is no further change in status. This is constant extrapolation.

Linear Extrapolation: Blender plots a straight line curve leaving and entering the curve. The action of the actor before and after first and last Keyframes follows these straight line curves.

Cyclic Extrapolation: Blender copies the graph between the first and the last Keyframes and duplicates it to infinity on either side of the graph. You **Make** the Extrapolation Cyclic or you **Clear** the Cyclic Extrapolation.



Constant Extrapolation



Linear Extrapolation

Cyclic Extrapolation

Figure 20.46

20.23 The Curve Properties Panel

The **Curve Properties panel** (Figures 20.47, 20.48) provides data and gives control to certain functions in the Graph panel. With the **Mouse Cursor** in the **Graph Editor**, press the **N key** to display the **Curve Properties panel** at the RHS of the Editor. The panel is divided into three Tabs; **F-Curve**, **Modifiers** and **View** Properties.

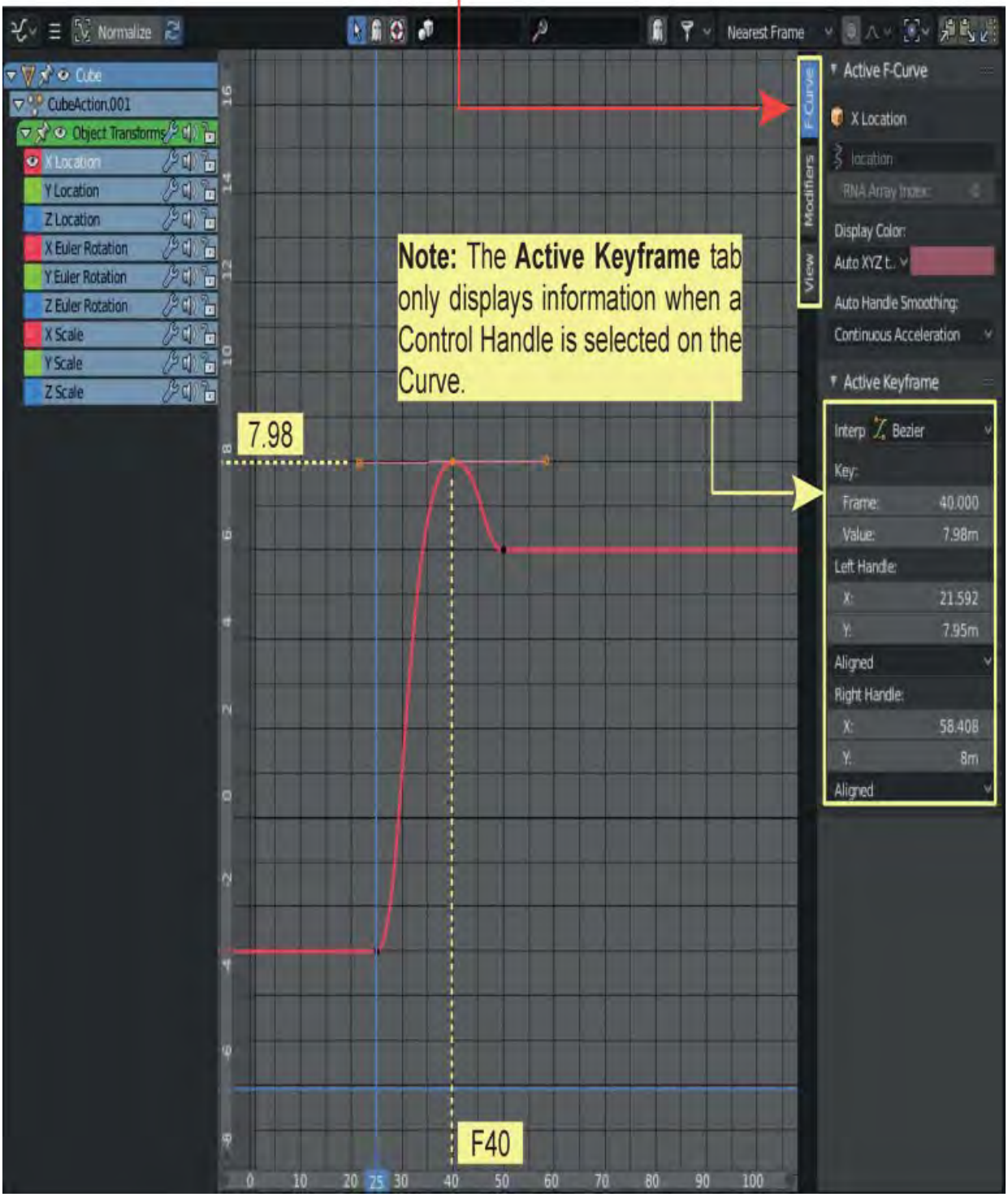


Figure 20.47

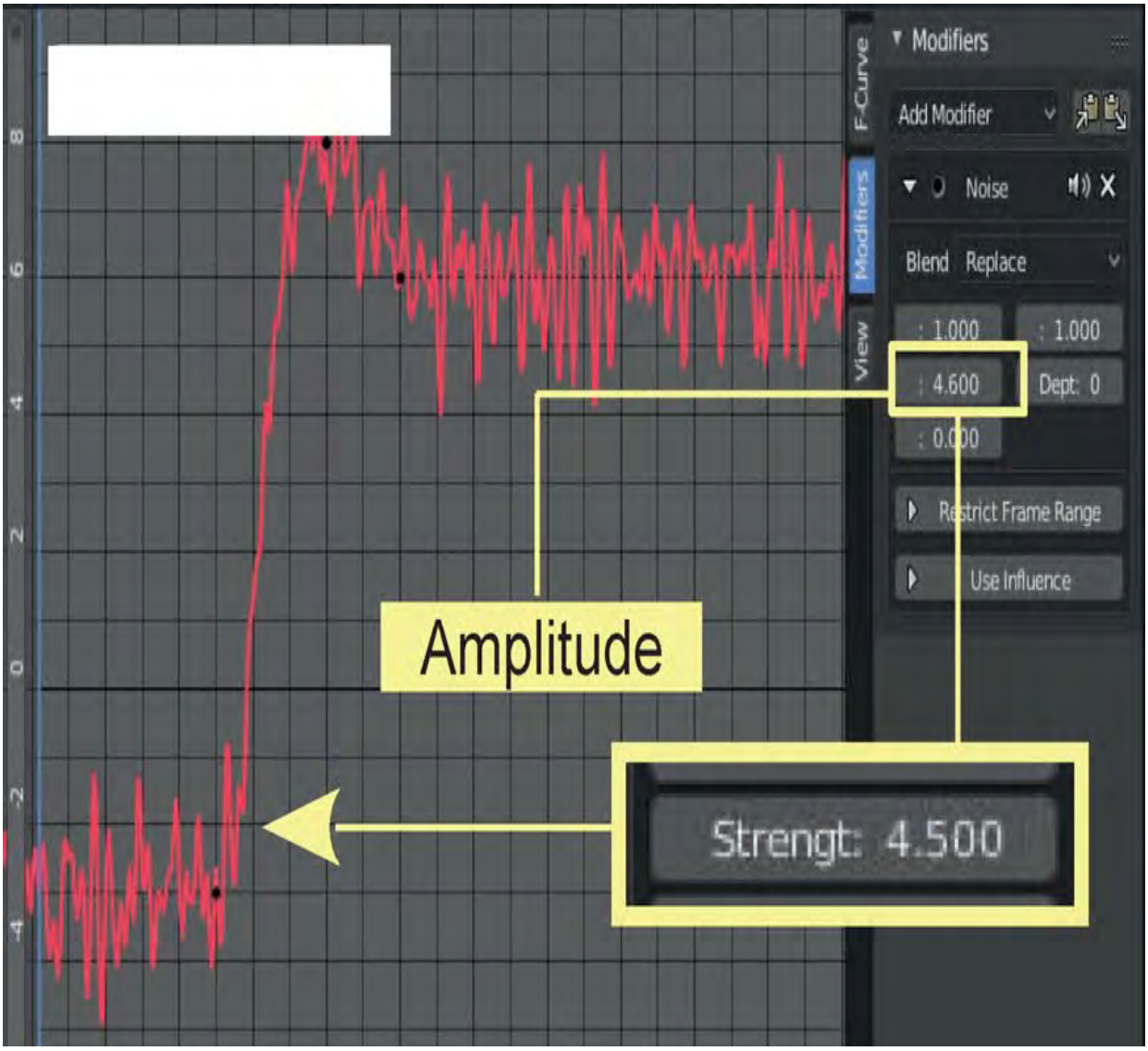


Figure 20.48

This introduction to the Curve Properties panel is presented to make you aware of its existence. Experiment with the values, especially the **Modifiers.** For example: With only the X Location Curve displayed in the Graph Editor click on the **Modifiers Tab** and then click Add Modifier. Select the **Noise Modifier** which adds a jittered effect to the Curve. Playing the animation sees

the Cube shake as it moves in the 3D Viewport Editor. For a more dramatic shake increase the **Amplitude** of the **Noise.**

20.24 Animating Rotation

Animating Rotation deserves special consideration when attempting to create a continuous smooth Rotation. As an example use a **UV Sphere Object** in **Top Orthographic View** with one Vertex moved to form a pointer (Figure 20.49). In Figure 20.49 a dial has been drawn showing Rotation graduations with the intended Rotation indicated.

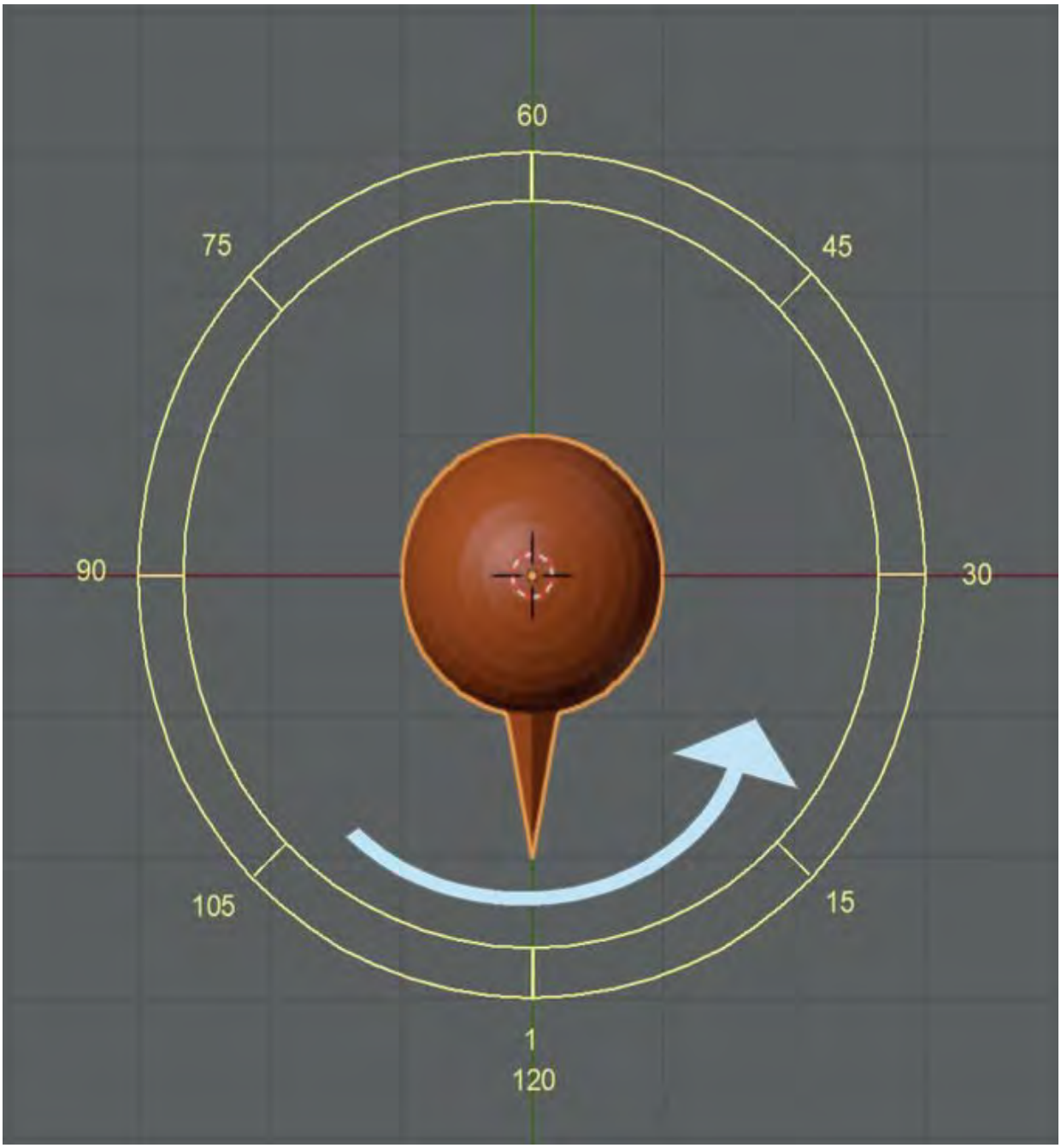


Figure 20.49

Divide the 3D Viewport Editor into three panels creating a **Graph Editor, 3D Viewport in Top Orthographic View** and a **Dope Sheet Editor.** The Timeline editor is retained across the bottom of the arrangement.

In the **Timeline Editor** turn on **Automatic Keyframing.**

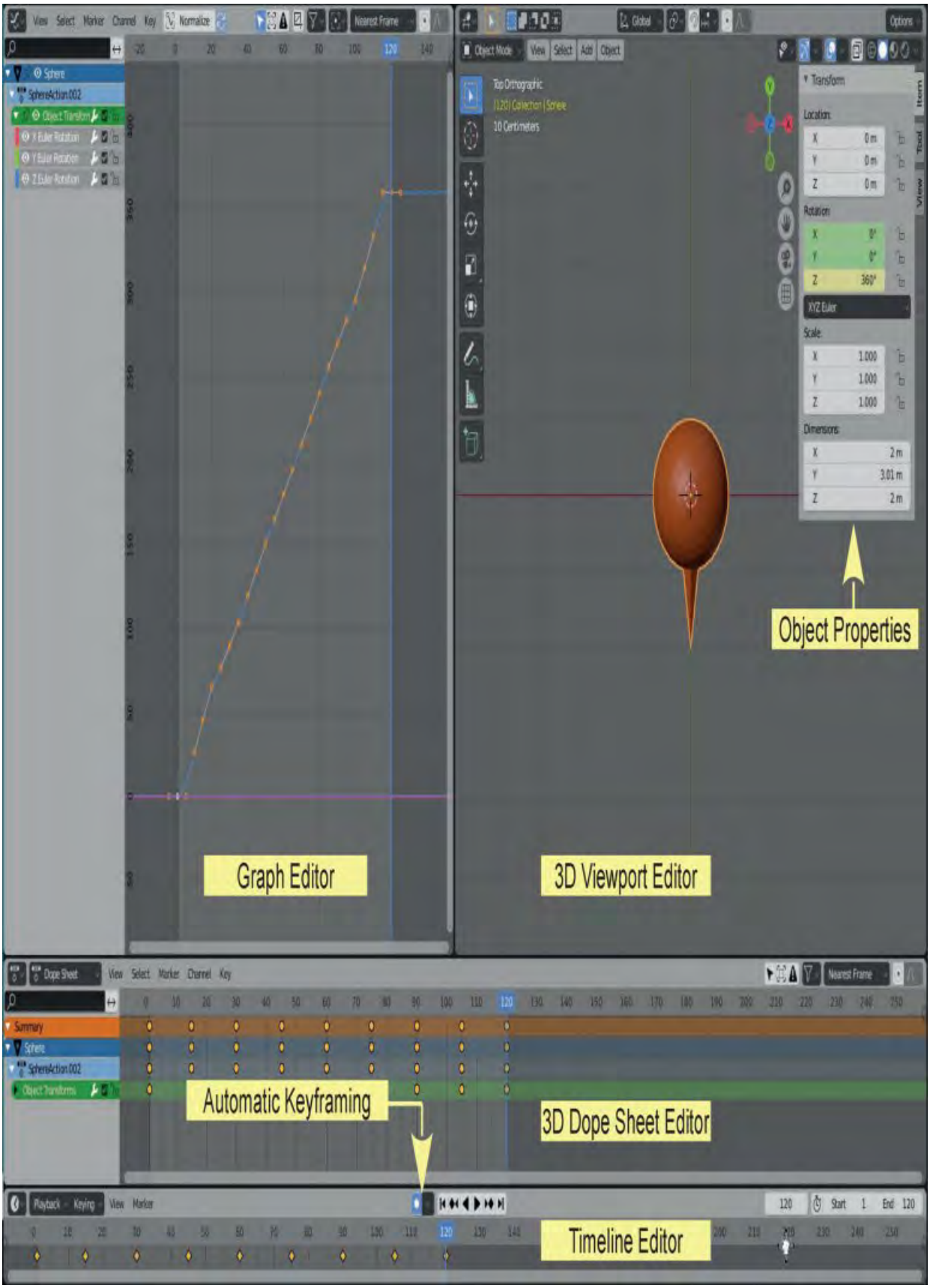

Graph Editor
3D Viewport Editor
Object Properties
Automatic Keyframing
3D Dope Sheet Editor
Timeline Editor

Figure 20.50

In the **3D Viewport Editor** have the modified UV Sphere selected and press the **N Key** to display the **Object Properties Panel.**

Note: In Figure 20.49 the dial graduations are expressed as **Keyframes in the Timeline.**

Rotation Procedure

With the Sphere selected and the pointer directed towards Frame 1 on the dial, have the **Dope Sheet Cursor** at Frame 1 in the Timeline. With the **Mouse Cursor** in the 3D Viewport Editor press the **I Key** and select **Rotation** as the Keyframe Type. Rotation Keyframes are entered in the Dope Sheet Timeline at Frame 1. **Note:** This is manually inserting the first Keyframe.

Move the **Dope Sheet Cursor** to Frame 15. In the **Object Properties Panel,** for the Sphere, enter **Z Axis Rotation: 45°.** Keyframes are **automatically** inserted in the Timeline at Frame 15.

Move the Dope Sheet Cursor to Frame 30, enter Z Axis Rotation: 90 in the Object Properties Panel. Keyframes are automatically inserted in the Timeline at Frame 30.

Repeat the procedure all the way round to Frame 120 which coincides with frame 1 on the dial. Keyframes will have been inserted in the Dope Sheet Timeline at 15 Frame intervals from Frame 1 to Frame 120.

In the Graph Editor you see a graph line consisting of a series of segments punctuated by Control Handles (Figure 20.51). The Graph Line is a series of **Bezier Curves.** By zooming in on the Curve you will observe a slight lead in and lead out at each Handle. This is more pronounced at the beginning and end of the Graph. The lead in and lead out results in a hesitation in Rotation at each Handle.

90
80
70
60
50
40
30
20
10
0
-10
Lead In

Figure 20.51

With the default settings, when the Play button is pressed in the Timeline Editor the Sphere Rotates, hesitating as each Keyframe (Handle) is passed, from Frame 1 to frame 120 then stops.

The Dope Sheet and Timeline Cursors continue to move on to frame 250 then the Animation is repeated.

To produce a smooth continuous Rotation have the **Mouse Cursor** in the **Graph Editor,** then press **Shift + E Key** and select **Linear Extrapolation** in the menu that displays. Change the Animation End Frame to 120 in the Timeline Editor Header. With these changes a continuous smooth Rotation will be seen when the Animation is played.

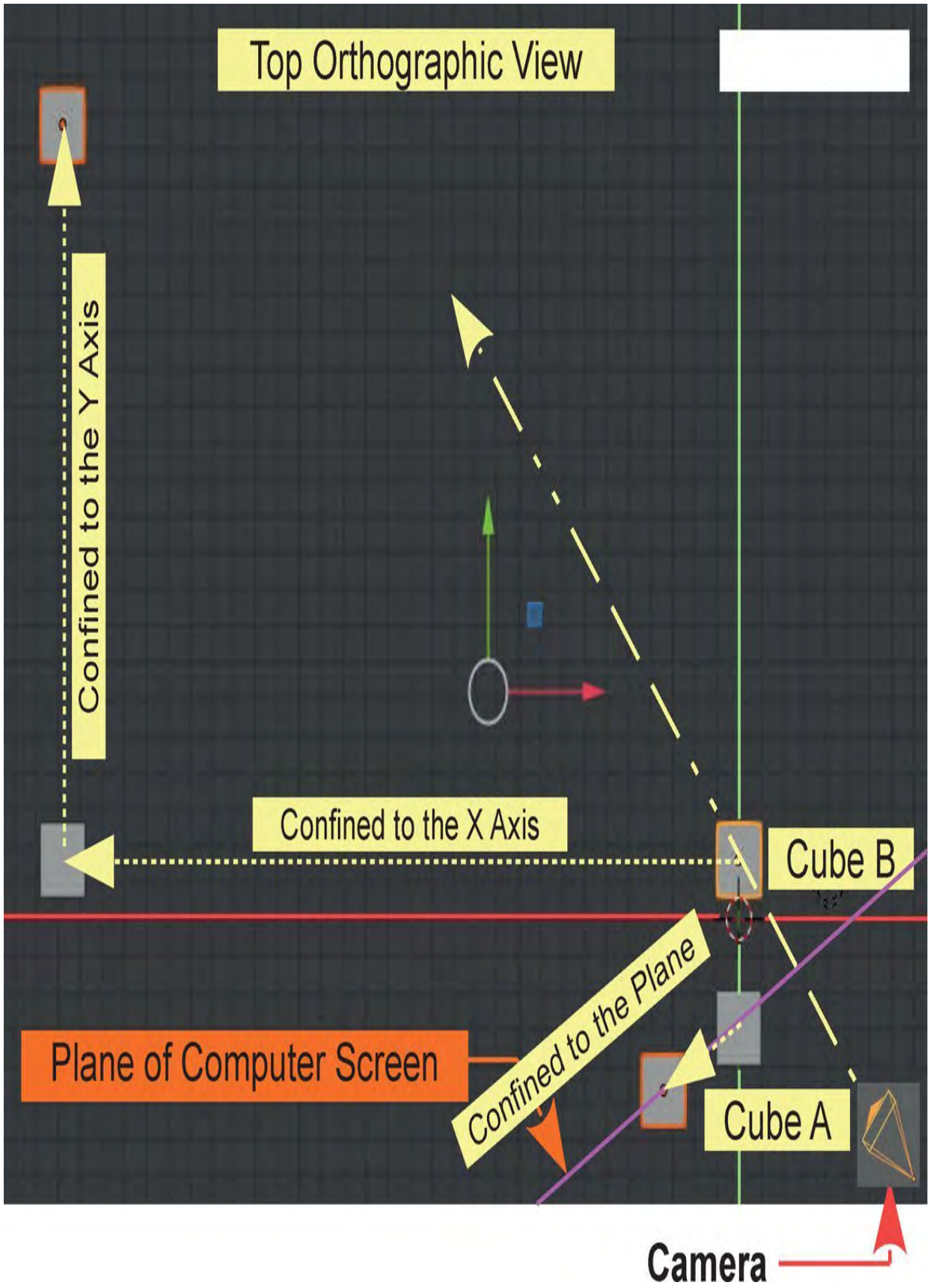

Top Orthographic View
Confined to the Y Axis
Confined to the X Axis
Cube B
Confined to the Plane
Plane of Computer Screen
Cube A
Camera

Figure 20.2

Linear Extrapolation

Entering **Liner Extrapolation** in the Graph Editor straightens out the curve and extends the curve to infinity. This has the effect of producing a smooth constant Rotation of the Object no matter what End Frame is set in the Timeline Editor.

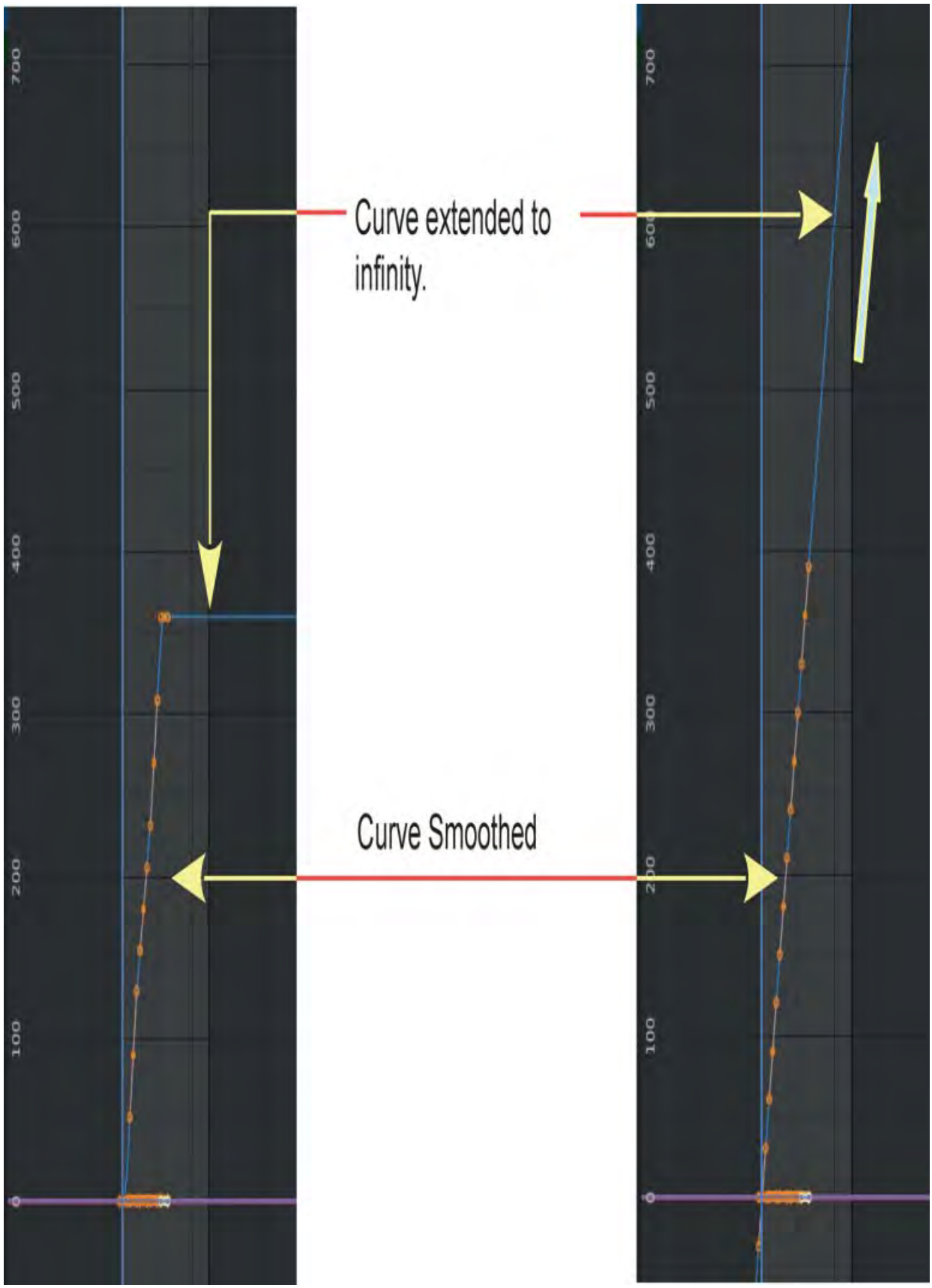

Curve extended to infinity.
Curve Smoothed

Figure 20.53

Speed of Rotation

The Speed of Rotation depends on the Frame Rate of the Animation playback set in the Properties Editor, Output Properties, Dimensions Tab. The default Rate is 24 fps (Frames per Second). The Speed of Rotation is also determined by the number of Frames between Keyframes. For one Revolution you have 120 Frames, therefore, 120 F / 24Fps = 5 Seconds per Revolution. To alter the speed **Scale the Timeline.** With the Mouse Cursor in the Timeline Editor, press the **S Key** and drag the Mouse.

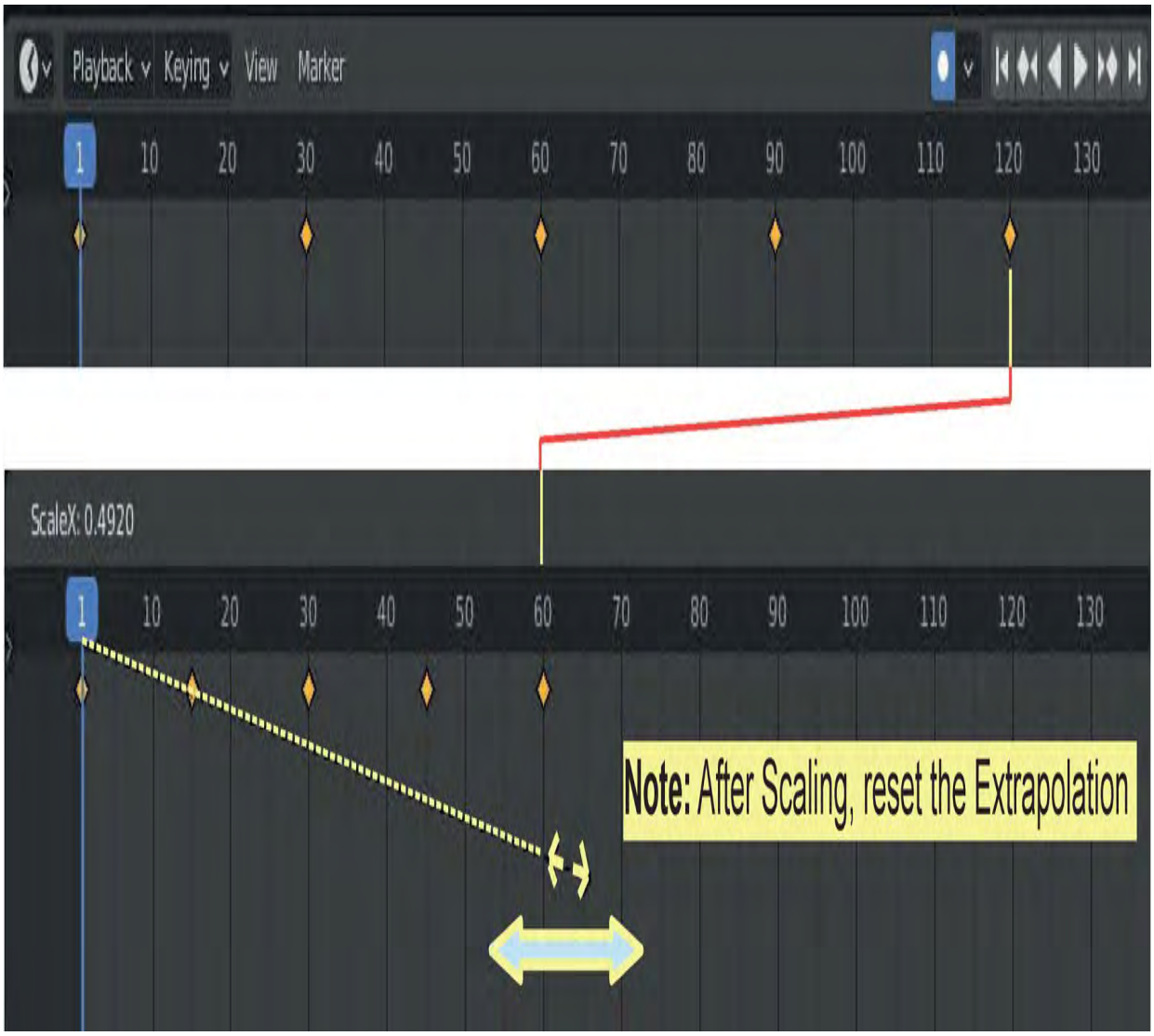


Figure 20.54

20.25 Rotation Using F-Curves

The Blender Manual states, “After animating some property in Blender using Keyframes you can edit their corresponding curves. When something is “animated”, it changes over time. This curve is shown as something called an **F-Curve.** Basically what an F-Curve does is an interpolation between two animated properties. In Blender, animating an object means changing one of its properties, such as the Object’s Location, or its Scale.

As demonstrated, when an Object is animated a Curve is created in the Graph Editor. It, therefore, follows that this is an **F-Curve.** When considering the default Cube Object in the 3D View Editor, before it is animated, there is no Curve in the Graph Editor. The Cube may be made to Rotate by animating to create a **Location Curve** then editing the Curve using Modifiers.

Animate the Cube to move from minus 5 units on the XAxis to plus 5 units on the XAxis between Frame 1 and frame 180. An **F-Curve** is drawn in the Graph Editor.

Normalize
Nearest Frame
Cube
CubeAction
Object Transforms
X Location
Y Location
Z Location
F-Curve for the X Location Channel
Y Location Curve
With the Mouse Cursor in the Graph Editor press the N Key to display the Properties Panel
Active F-Curve
X Location
location
RNA Array Index: 0
Display Color:
Auto XYZ to ..
Auto Handle Smoothing:
Continuous Acceleration
Active Keyframe
Interp.. Bezier
Key:
Frame: 1.000
Value: -5m
Left Handle:
X: -58.667
Y: -5m
Auto Clamped
Right Handle:
X: 60.667
Y: -5m
Auto Clamped
F-Curve
Modifiers
View

Figure 20.55

With the Curve drawn, playing the animation shows the Cube move from minus 5 to plus 5 along the X Axis in the 3D View Editor. As well as the X Location Curve a Y Location Curve has also been created (straight line).

Both Curves may be modified to produce a Rotation of the Cube.

In the **Properties Panel** click on the **Modifiers Tab,** click **Add Modifier** and select **Built in Function** (Figure 20.56). The default Built in Function is **Sine** which immediately changes the Bezier Curve in the Graph Editor to a Sine Curve or rather a Sinusoidal Curve since it repeats to infinity (Figure 20.57 over).

Figure 20.56

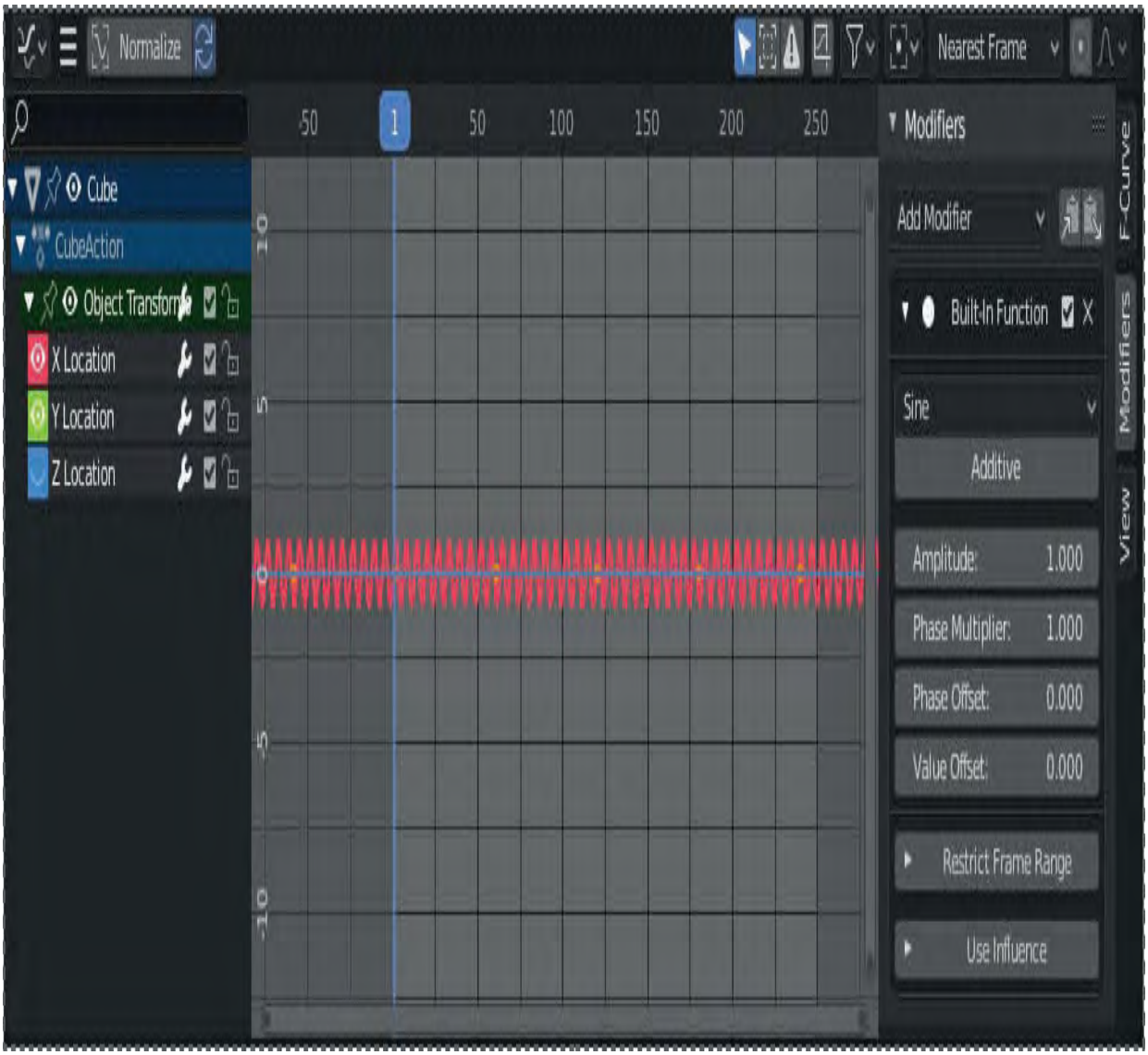


Figure 20.57

Playing the animation shows the Cube in the 3D Viewport Editor oscillate on the X Axis. The degree of oscillation is governed by the Amplitude value in the Modifier. The movement of the Cube is best seen in Top Orthographic View.

In the Modifier panel, increase the Amplitude to 2.000 and reduce the Phase Modifier to 0.100. This reduces the oscillation of the Cube to a nice smooth motion

Figure 20.58

The Curve under consideration at this point has been the Curve representing the **X Axis** motion of the Cube (see the **X Location Channel** in the Graph Editor). As previously stated a **Y Axis** motion Curve is also drawn (straight line) as seen by selecting the **Y Location Channel.** By applying a Built in Function Modifier and making it **Type Cosine** with Amplitude 2.00 and Phase Multiplier 0.100, the Cube rotates around the center of the Scene.

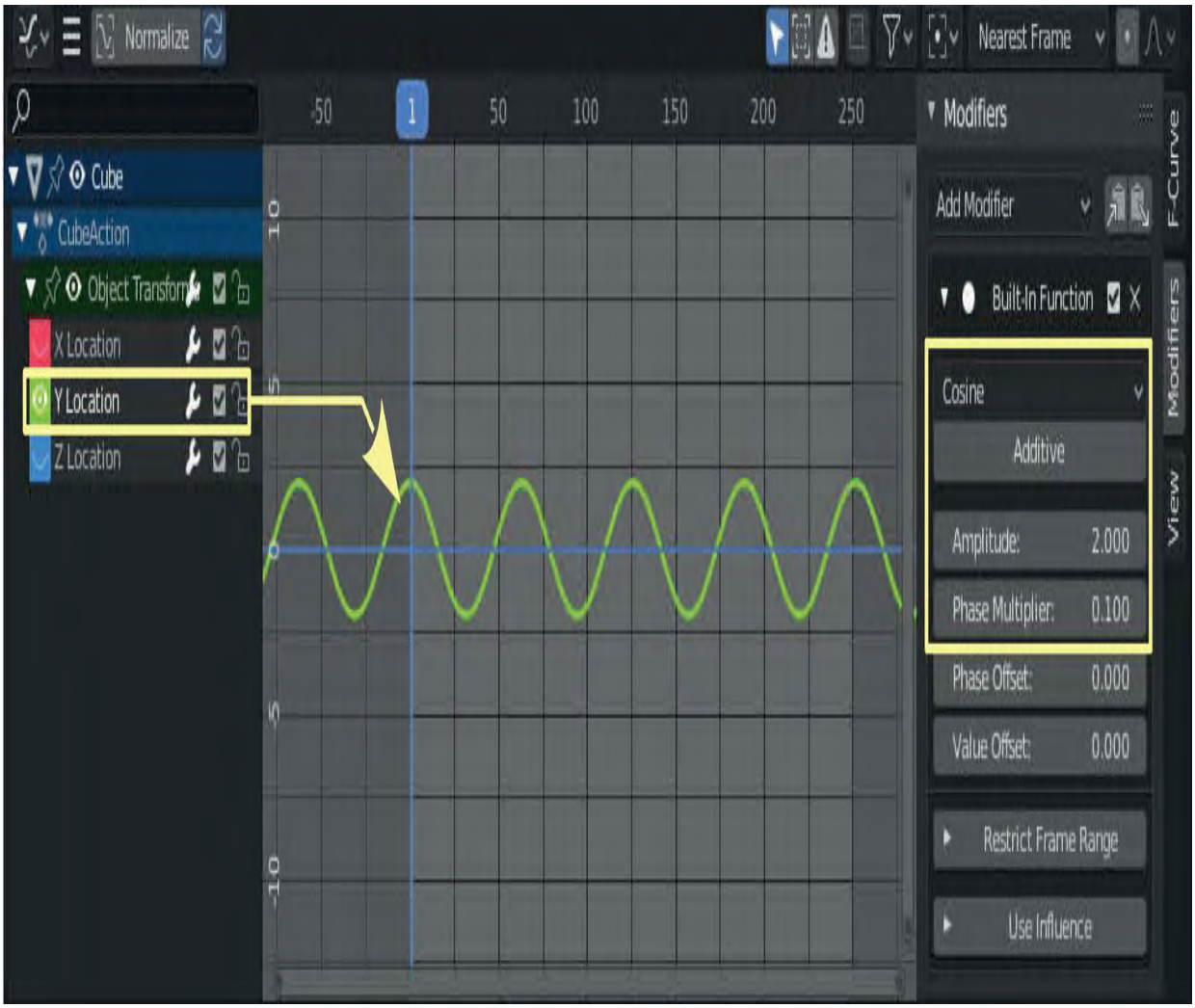


Figure 20.59

20.26 Animating Other Features

There are many features in Blender which may be animated. For example:

Material (colon Animatior

As an example of animating color change perform the following using the Animation Workspace;

Add a UV Sphere in the 3D Viewport Editor and set the surface to Smooth Shading. In the Properties Editor, Material buttons, add a Material and leave Use Nodes active.

Change the 3D Viewport Editor to rendered Viewport Shading.

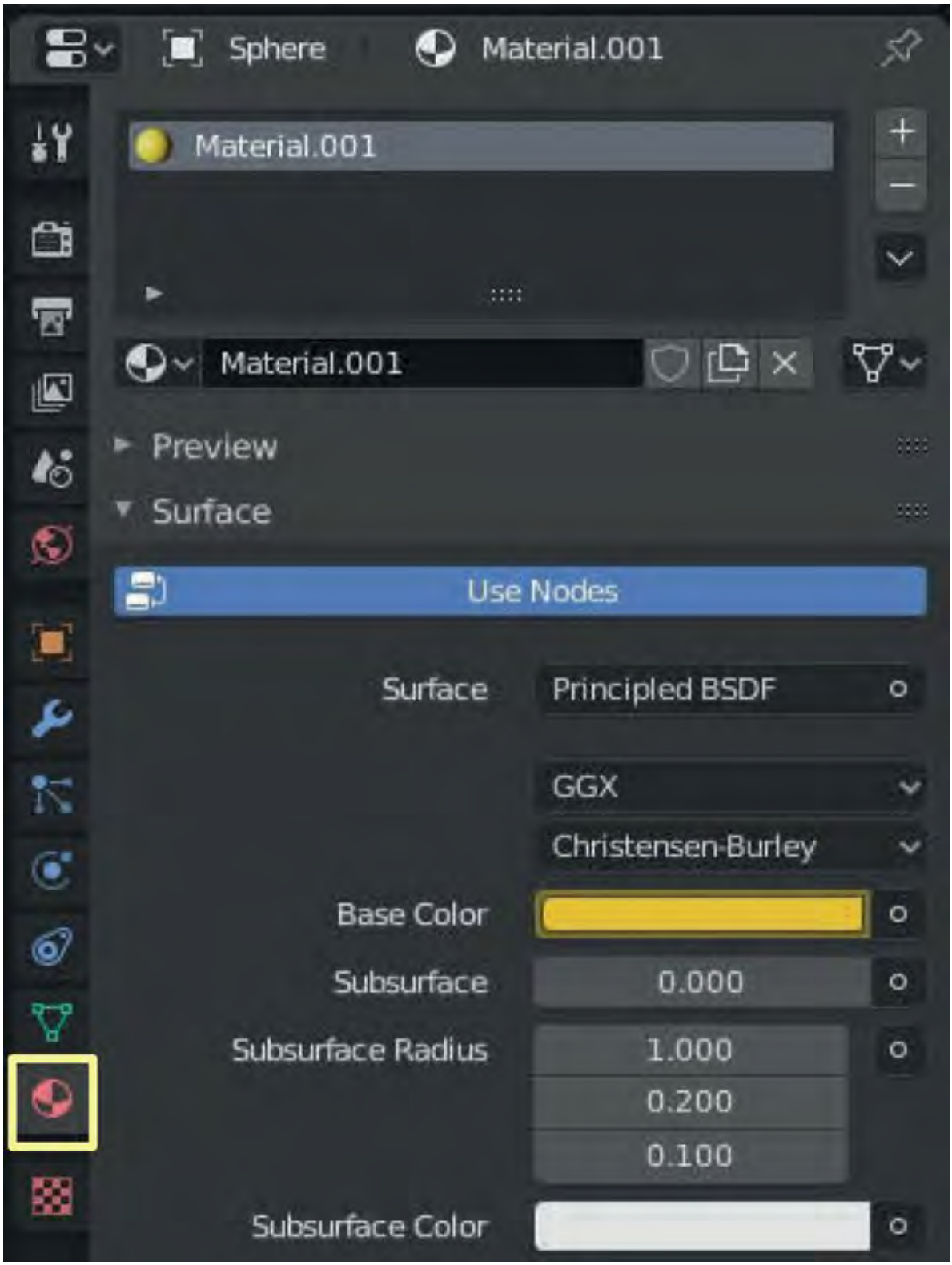

Sphere
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
1.000
0.200
0.100
Subsurface Color

Figure 20.60

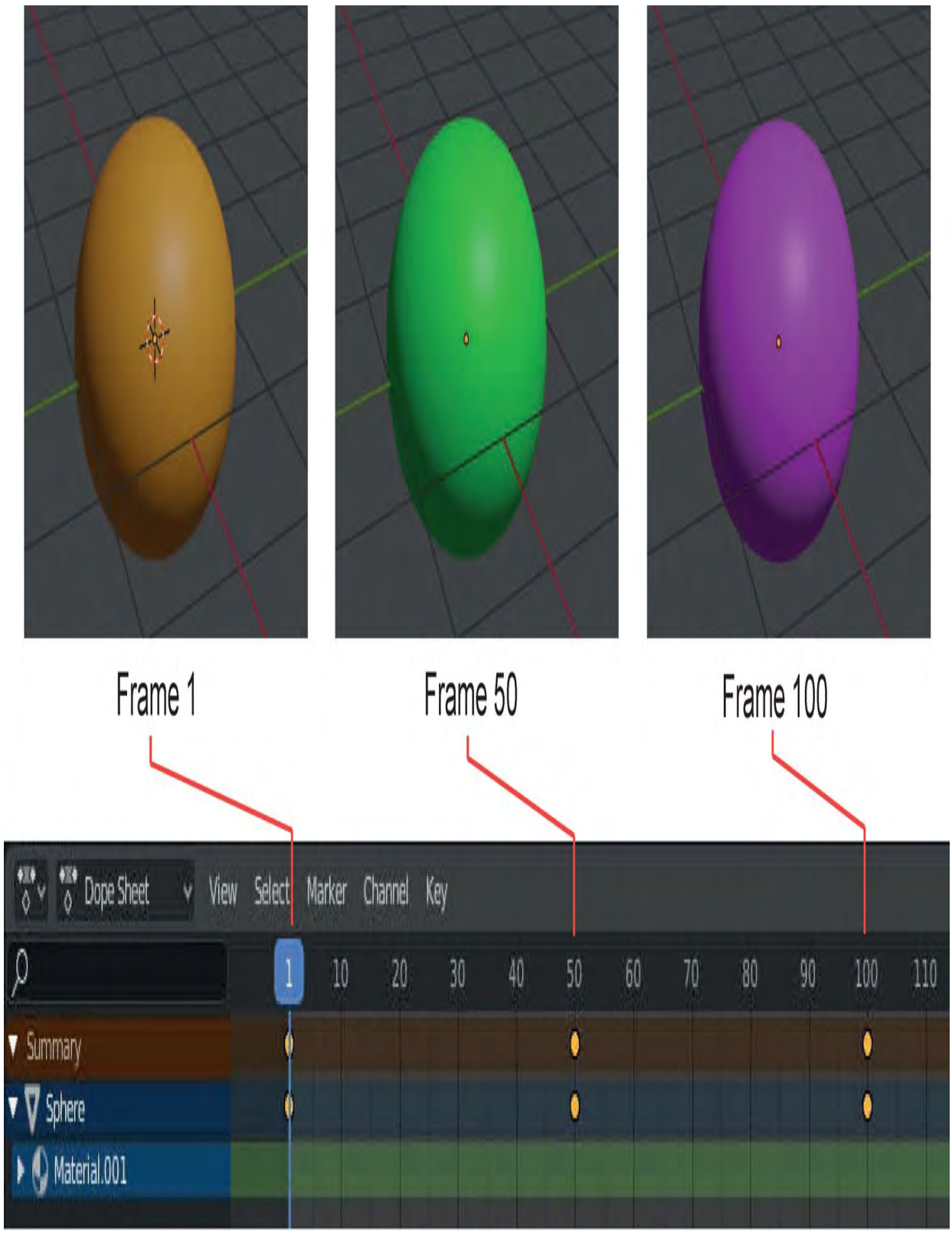
Figure 20.61

In the Material buttons, click on the base Color bar and select a color. **RMB click the color bar** and select **Insert Keyframe** from the menu.

This enters a Keyframe in the Dope Sheet Timeline at the location of the Cursor (the default is frame 1).

Relocate the Timeline Cursor to another Frame (Frame 50). Select a different color. RMB click the color bar and select Insert Keyframe. A Keyframe is entered at frame 50.

Repeat for Frame 100 then play the animation to see the color change in the 3D Viewport Editor.



Timeline Cursor at Frame 1 (blue line)

Figure 20.62

The size of a Spotlight beam may be animated to change.

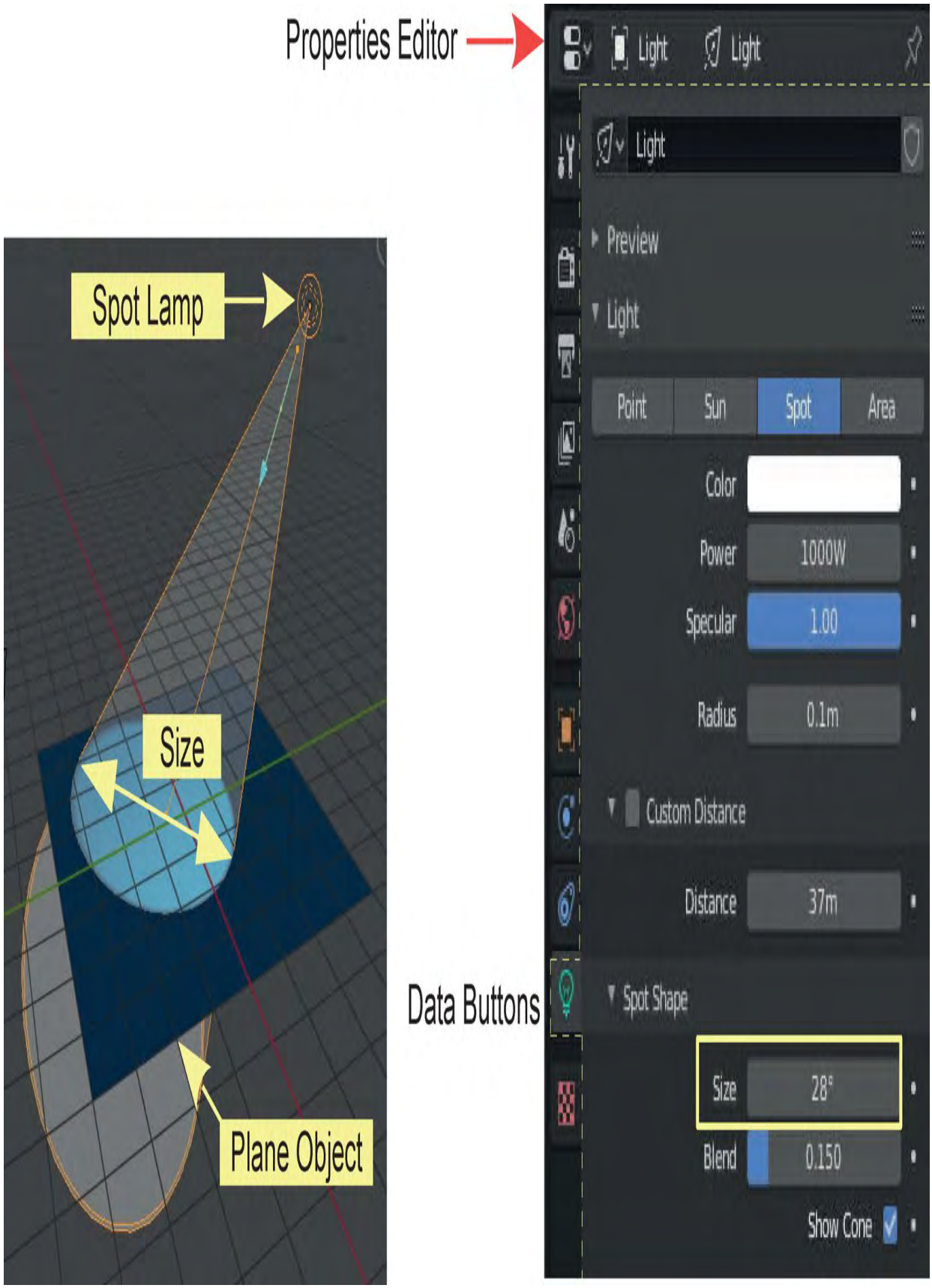

Properties Editor
Spot Lamp
Size
Plane Object
Data Buttons
Light
Light
Light
Preview
Light
Point
Sun
Spot
Area
Color
Power
1000W
Specular
1.00
Radius
0.1m
Custom Distance
Distance
37m
Spot Shape
Size
28°
Blend
0.150
Show Cone

Figure 20.63

Using the 3D Animation Workspace be at Frame 1 in the Timeline. With the Spot Lamp selected in the 3D Viewport Editor, in the Properties Editor, Data buttons, Spot Shape tab, **RMB click** on Size and select **Insert Keyframe.** Change to Frame 50. Change the Size and insert a second Keyframe. Repeat at frame 100 and play the animation.

20.27 Keying Sets

In a basic capacity **Keying Sets** provide the means of assigning multiple **Animation** properties to an Object, then creating all the Keyframes in a single go.

To demonstrate have the default **Layout Workspace** with the **Timeline Editor** sized vertically, divided in two and having one half changed to the **Dope Sheet Editor.** The upper part of the Screen is the 3D Viewport Editor containing the default Cube. The Timeline and Dope Sheet Cursors will be located at Frame 1.

Keyframes will be created for the Location and Material color. Have the 3D Viewport Editor in **Material Preview Viewport Shading Mode.**

In the 3D View Editor move the Cube back along the X Axis. Note: The default Cube has the default gray Material applied.

With the Mouse Cursor in the 3D View Editor, press the **N Key** to display the **Object Properties Panel.**

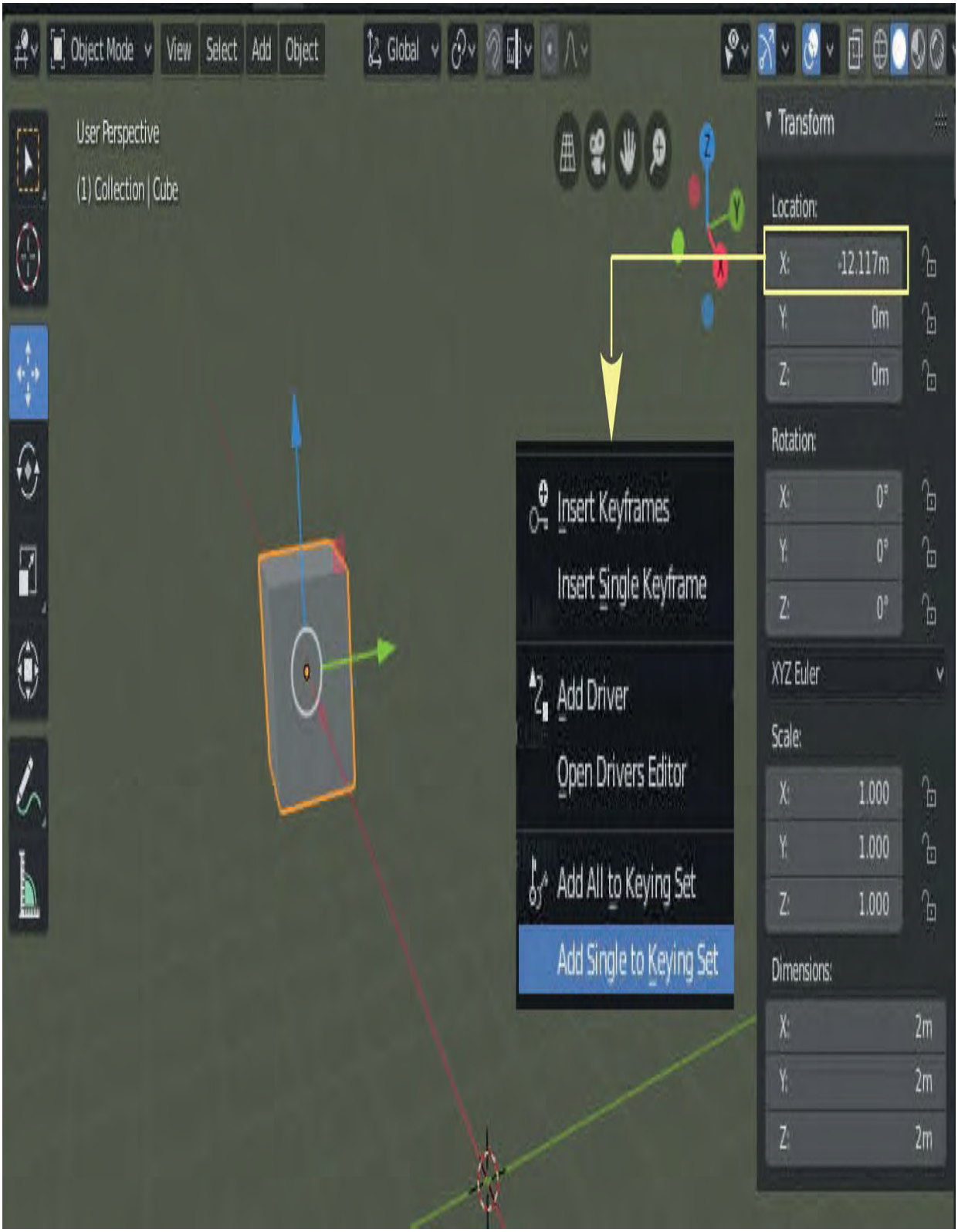

Object Mode
View
Select
Add
Object
Global
User Perspective
(1) Collection | Cube
Transform
Location:
X: -12.117m
Y: 0m
Z: 0m
Rotation:
X: 0°
Y: 0°
Z: 0°
XYZ Euler
Scale:
X: 1.000
Y: 1.000
Z: 1.000
Dimensions:
X: 2m
Y: 2m
Z: 2m
Insert Keyframes
Insert Single Keyframe
Add Driver
Open Drivers Editor
Add All to Keying Set
Add Single to Keying Set

Figure 20.64

Under **Location** in the panel, RMB click on **X Location** and select Add **Single to Keying Set.** This is saying you wish to add a single Keyframe to the Keying Set. It does not enter a Keyframe in either Timeline Editor.

RMB click on **Y Location** and repeat the above which adds a second Keyframe request to the set. Again, Keyframes are not entered in the Timelines.

Now request a Keyframe for the Material color. In the **Properties Editor, Material buttons, RMB** click on the **Base Color** bar and select **Add to Keying Set.**

Keyframes have been requested for the X and Y Location and the Material Color of the Cube. The Timeline and Dope Sheet editor Cursors are at Frame 1. The Keyframes are, therefore, requested for Frame 1.

In the **Timeline Editor Header** click on the **Keying** button. In the menu that displays, adjacent to **Button Keying Set** click on the Key button **with the plus sign above the key icon.** This inserts the requested Keyframes in the Timelines at Frame 1. A message displays in the Header at the bottom of the Screen.

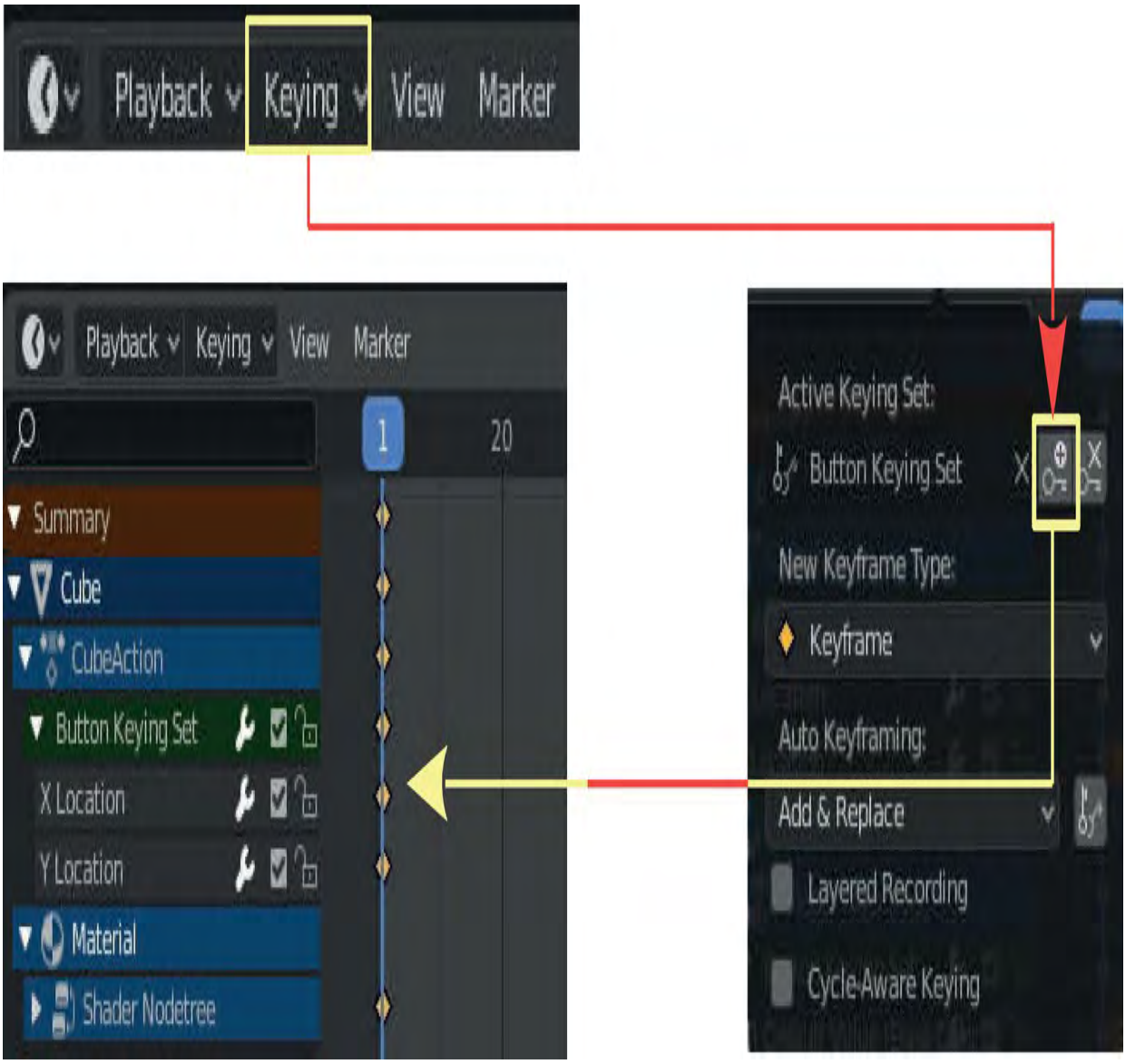


Figure 20.65

Move the Timeline Cursors to Frame 50. In the 3D Viewport Editor relocate the Cube on the XY Plane. In the Object Properties Panel, RMB click on the X and Y Location channels and request Keyframes. Change the Material base Color and request a Keyframe.

Repeat the Keying procedure, entering Keyframes in the Timelines.

Move the Timeline Cursors to a different Frame and repeat the requesting and adding Keyframes.

Scrubbing the Cursors in the Timelines shows the Cube moving and changing color.

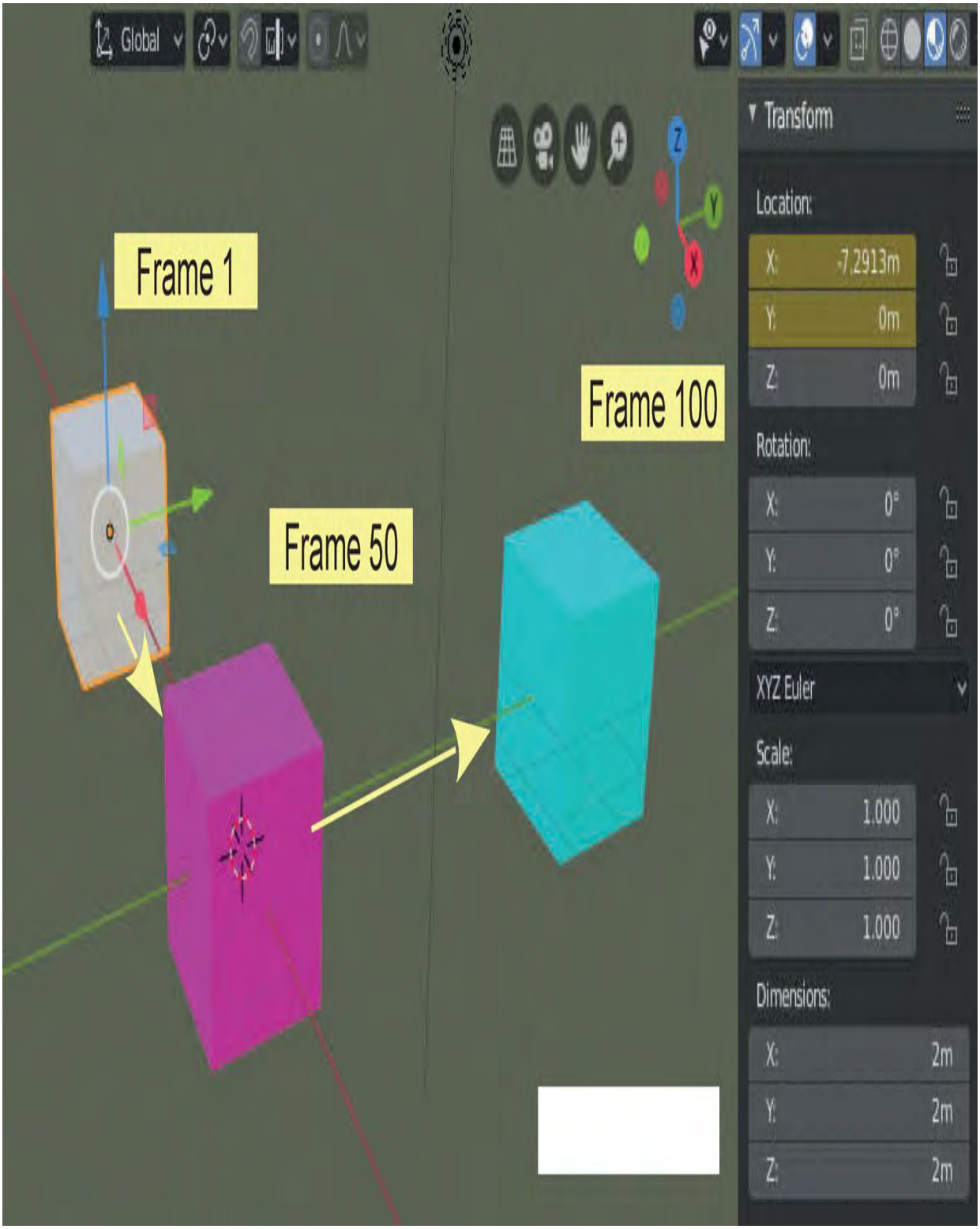


Figure 20.66

20.28 Animation Follow Path

As previously demonstrated, animating an Object to move is achieved by entering Keyframes in the Timeline with the Object located at positions in the 3D Viewport Editor. With the animation created it may be edited by modifying the Curve in the Graph Editor or relocating Keyframes in the Dope Sheet Editor Timeline.

On occasion you may wish to animate an Object to follow a pre-constructed Path. Once the Object is made to follow the Path, you reshape the Path to alter the movement of the Object in the 3D Viewport Editor.

To demonstrate the procedure for animation following a Path, open the **Animation Workspace and change the left hand** 3D Viewport Editor Camera View to the **Graph Editor.** Have the right hand 3D Viewport Editor in top Orthographic View with default Cube Object **at the center of the Scene.**

Deselect the Cube and add a **Curve Type: Bezier** to the Scene. You may use any Curve Type but the Bezier Curve has a nice profile when entered. Scale the Bezier Curve up six times (Figure 20.65).

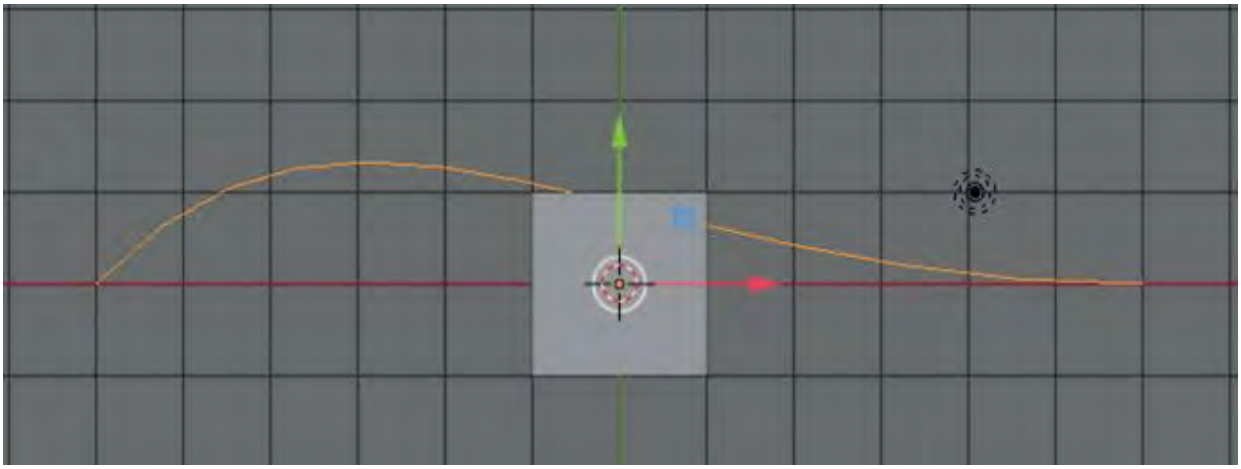


Figure 20.67

Deselect the Curve and **select the Cube.**

In the **Properties Editor, Object Constraints Properties,** click **Add Object Constraint** and select **Follow Path** in the Relationship category. The

Follow Path Tab displays (Figure 20.68).

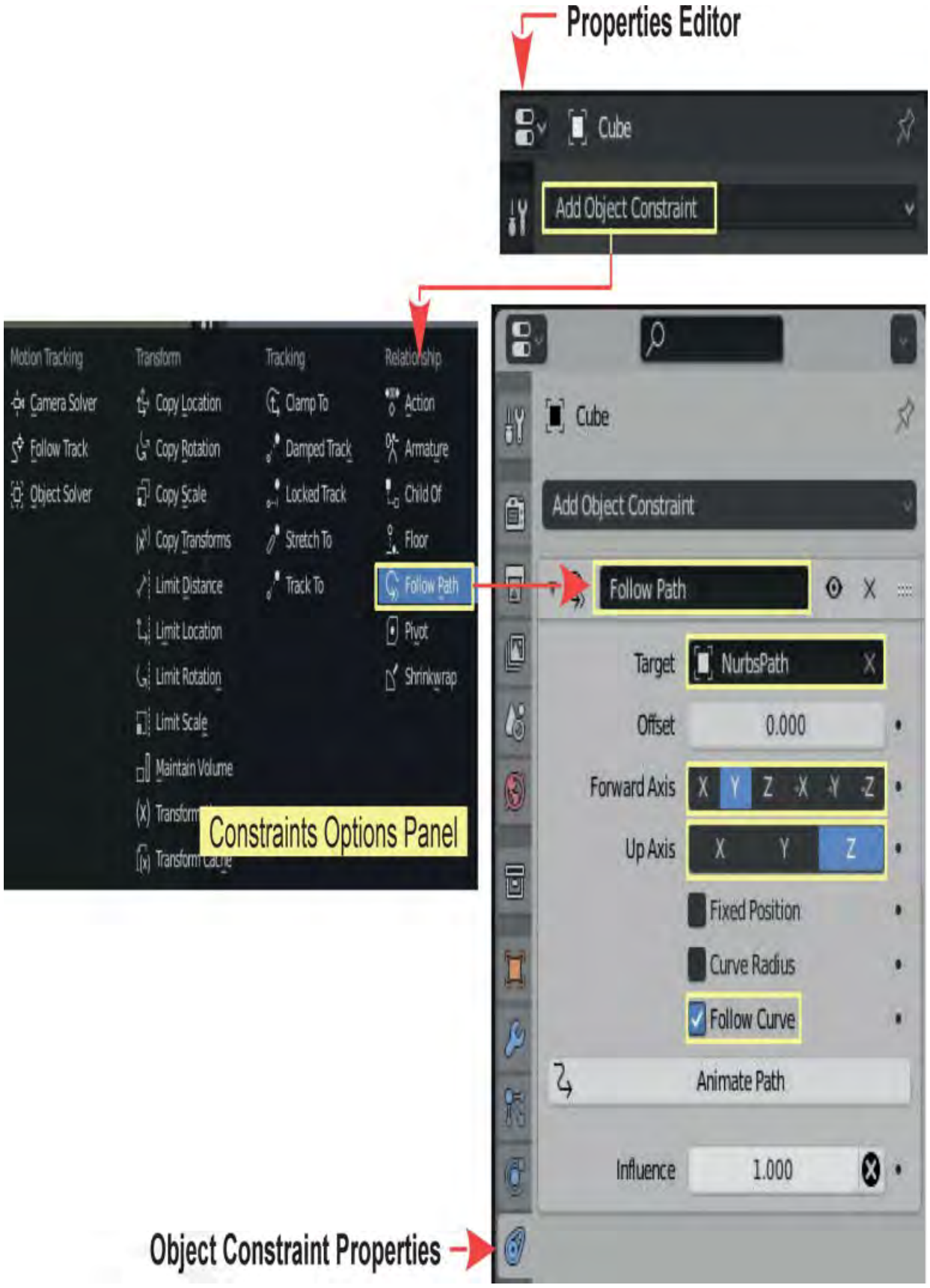

Properties Editor
Cube
Add Object Constraint
Motion Tracking
Camera Solver
Follow Track
Object Solver
Transform
Copy Location
Copy Rotation
Copy Scale
Copy Transforms
Limit Distance
Limit Location
Limit Rotation
Limit Scale
Maintain Volume
Tracking
Clamp To
Damped Track
Locked Track
Stretch To
Track To
Relationship
Action
Armature
Child Of
Floor
Follow Path
Pivot
Shrinkwrap
Constraints Options Panel
Cube
Add Object Constraint
Follow Path
Target
NurbsPath
Offset
0.000
Forward Axis
Up Axis
Fixed Position
Curve Radius
Follow Curve
Animate Path
Influence
1.000
Object Constraint Properties

Figure 20.68

In the **Follow Path Constraint** click in the bar where you see the **Target** icon and select **Bezier Curve** from the panel that displays. Also, check (tick) **Follow Curve. Note: Forward Y** is selected (the Object's Axis which points in the direction of movement).

Entering the Target locates the Cube in the 3D Viewport Editor at the **LH** end of the Curve Path. Checking **Follow Curve** aligns the Cube's Axis to the Path (Figure 20.69).

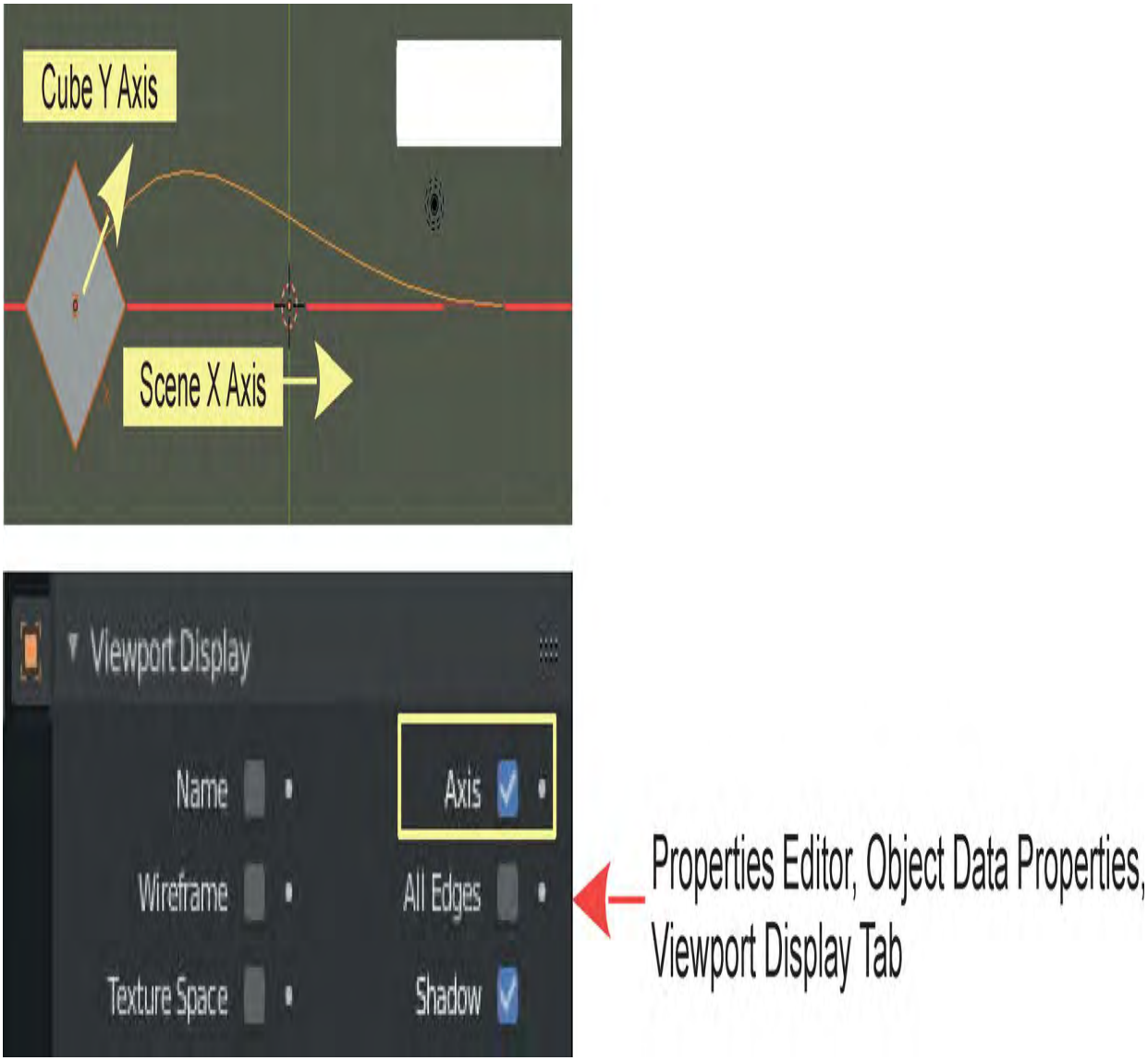


Figure 20.69

With the Cube aligned deselect the Cube and select the Curve Path.

Go to the **Properties Editor, Object Data Properties** (for the Curve Path) and see the **Path Animation Tab** at the bottom of the panel. Ensure that **Path Animation** is checked (Figure 20.70). Select the Cube and go back to the Follow Path Constraint. In the Constraint panel click **Animate Path.** Press Play in the Timeline Editor Header to see the Cube traverse the length of the Path.

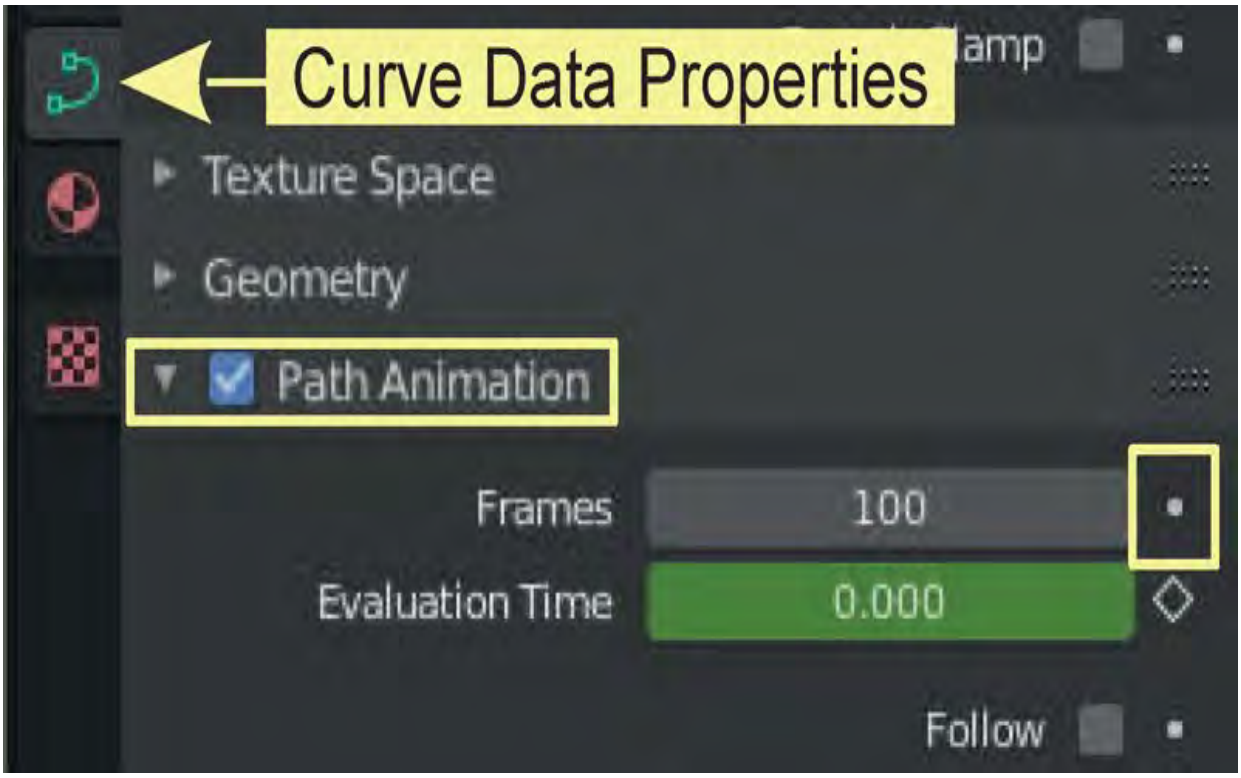


Figure 20.70

Note: At this point Keyframes have not been entered in the Timeline Editor.

You may select the Curve Path, Tab into Edit Mode and modify the shape of the Curve. If you Extrude the Path from either end, the Cube always locates at

the start of the Path at Frame 1 in the Animation. In Edit Mode the Curve Path displays with chevrons along its length indicating the direction of travel.

How quickly the Cube appears to traverse the length of the Curve Path in the 3D View Editor depends on the physical length of the Curve and the Frame Rate of the Animation (see the Properties Editor, Output buttons, Frame Rate).

Altering the Frame Rate will change the speed of movement along the Path but the Frame Rate is set for the output required in the video format being produced. **Do not change this.** There are two alternatives; Modifying Evaluation Time and Keyframing.

20.29 Evaluation Time

Evaluation Time settings are found in the Properties Editor, Data buttons, Path Animation Tab. The setting is defined by Blender as;

Evaluation Time.

Parametric position along the length of the curve that Objects ‘following’ it should be at (position is evaluated by dividing by the ‘ Path Length’ value).
In a practical sense you could consider the Evaluation Time as a percentage. At 0.000 % the Object is at the start of the Path, at 100.000 % it is at the end of the Path.

3D View Editor

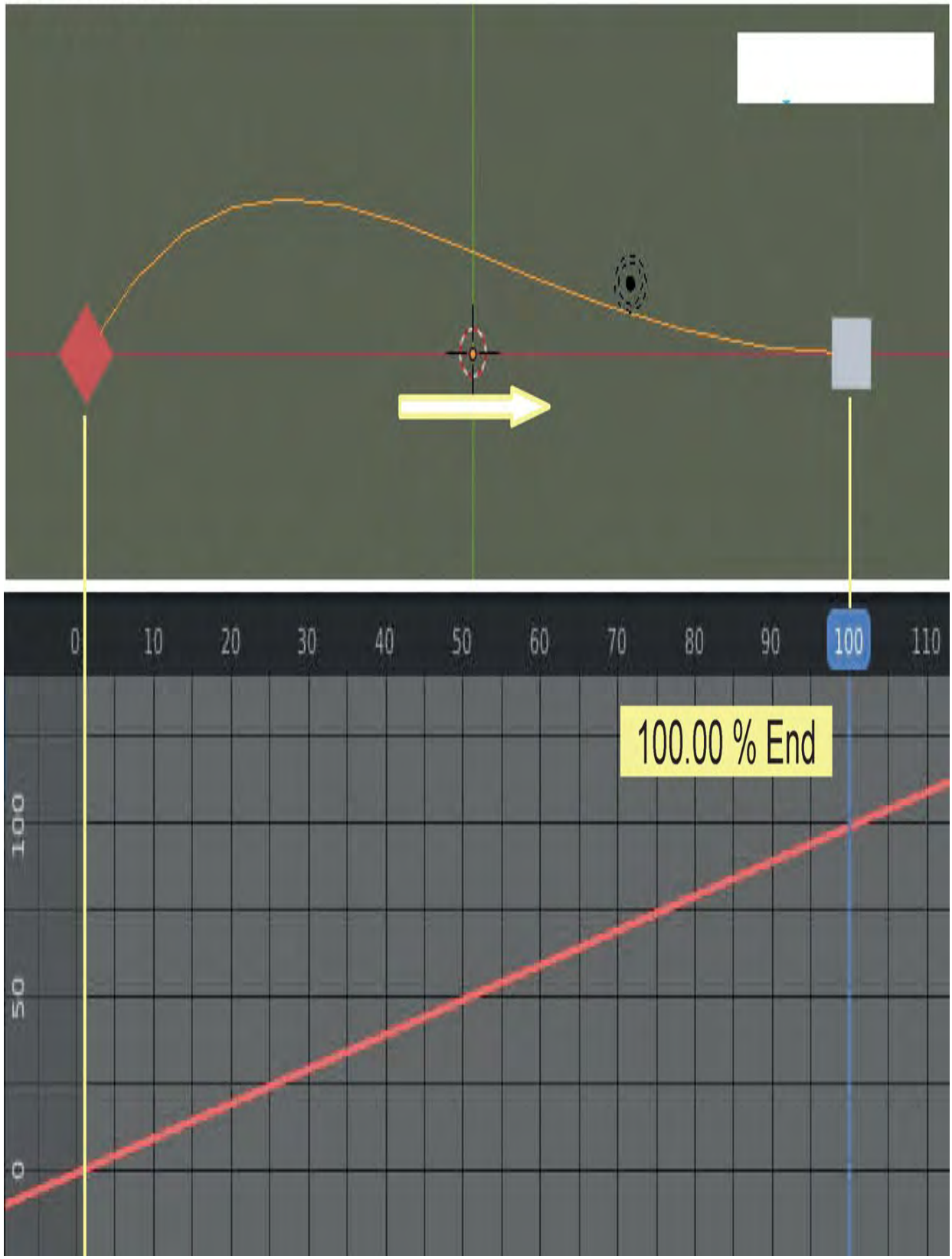


Figure 20.71

Graph Editor

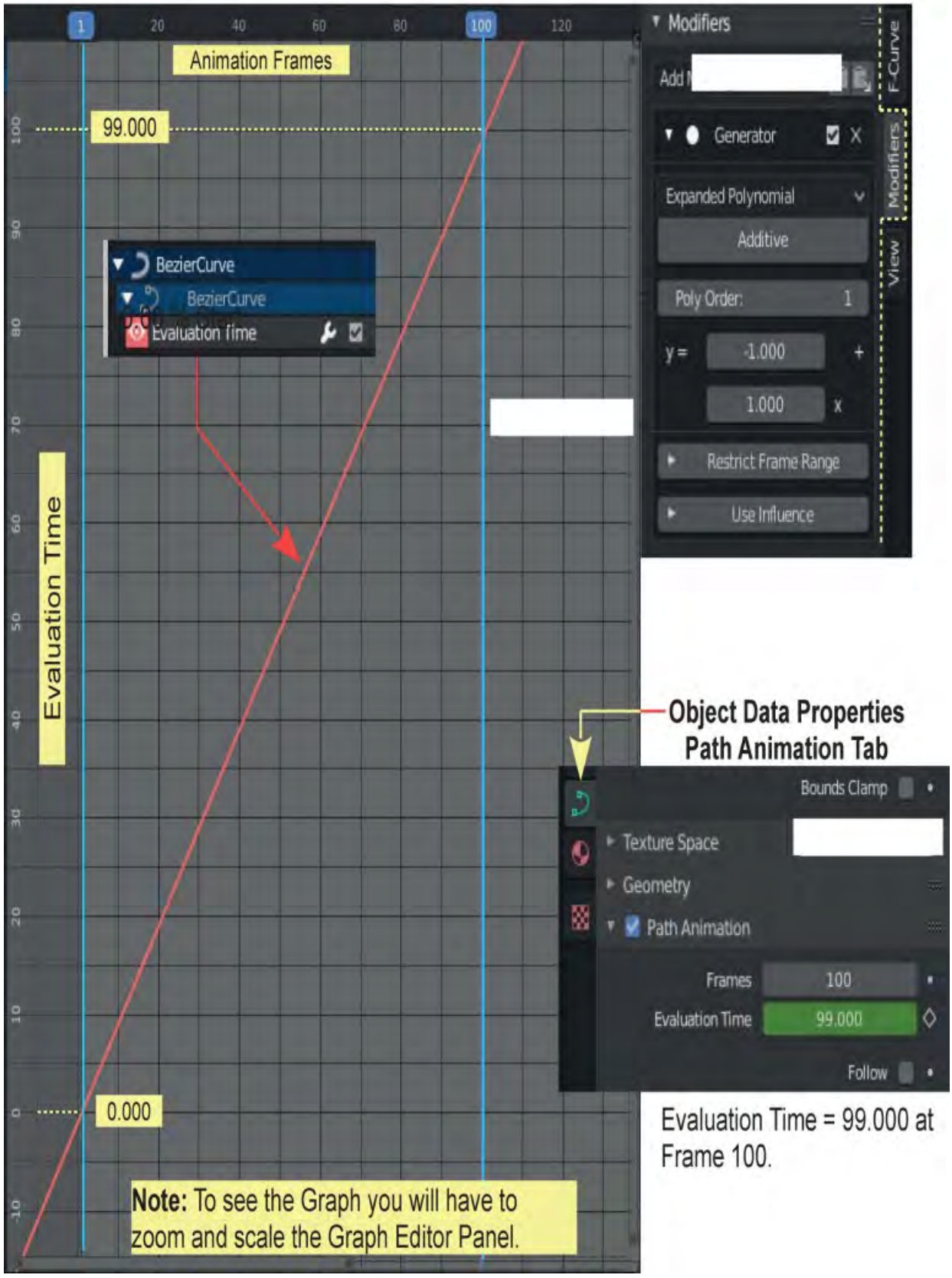


Figure 20.72
Figure 20.73
Figure 20.74

Graph Editor

Figure 20.69 shows a Cube Object scaled down 0.5 and a Bezier Curve Path scaled up 10 times. The Cube has a Follow Path Constraint applied with Target: BezierCurve. With the Bezier Curve Path selected, the Properties Editor, Object Data Properties, Path Animation Tab (Figure 20.65) shows the value; Frames 100. This is the default value which means the Cube will traverse the Path in 100 Frames of the animation. With the Graph Editor Cursor at Frame 1 the Evaluation Time is 0.000. With Cursor at frame 100 the Evaluation Time is 99.000 (Frame 100 — Frame 1 = 99). The default Frame Rate is 24 Frames per second, therefore, actual time to traverse the Path will be approximately 4 seconds.

The Graph Editor (Figure 20.73) shows the Evaluation Time Graph (red line) indicating these values. The Bezier Curve Path must be selected in the 3D Viewport Editor to see the Graph.

With the Mouse Cursor in the Graph Editor, press the N key to display the Properties Panel (Figure 20.75) and select the Modifiers Tab at the RHS of the panel. Note the two values;

Figure 20.75

y= -1.000 and 1.000 x

Increasing or decreasing **y = -1.000** shifts the Graph Line to the right and left (the slope of the graph remains the same). Altering the **1.000 x** value changes the slope of the Graph.

Altering either value alters the position of the Cube on the Curve Path at a particular Frame thus affecting the speed of traversing.

20.30 Keyframing

Up to this point **Keyframes** have not been entered in the **Timeline.** The Graph Editor only displays the Evaluation Time (when the Curve Path is selected in the 3D Viewport Editor). Entering Keyframes in the Dope Sheet Timeline allows you to manipulate where the Cube is located along the Curve Path at a particular Frame in the animation. This also affects the speed of traversing the Path.

With the Cube path arrangement set up as shown in Figure 20.69 and with the Timeline Cursor at Frame 1, add a **Follow Path Constraint** to the Cube with **Target: BezierCurve.**

Do NOT Animate the Path.

In the Path Animation Tab the Frames bar turns yellow and the button becomes a diamond shape indicating Keyframes have been added (Figure 20.7).

Move the Timeline Cursor to Frame 100, the end of the Path as indicated by Frames: 100.

Note: Frame 100 is the default value. You may reset this, bearing in mind, the default TOTAL animation length set in the Timeline is 250 Frames (Start: 1, End: 250).

Change the Evaluation Time to 100 (or to the value you have set).

Press the diamond button to enter Keyframes at Frame 100 in the Timeline (Figure 20.76 — 77).

Select the Bezier Curve Path and in the Properties Editor, Object data buttons, Path Animation Tab, click the button at the end of the Frames bar. **Keyframes** are entered in the Timeline.

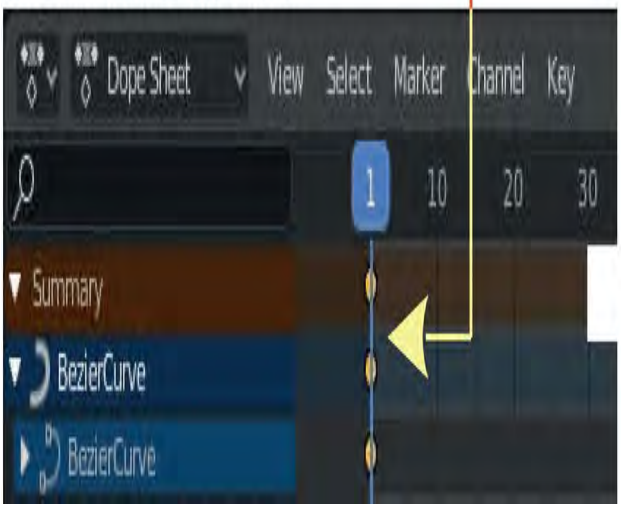


Figure 20.76

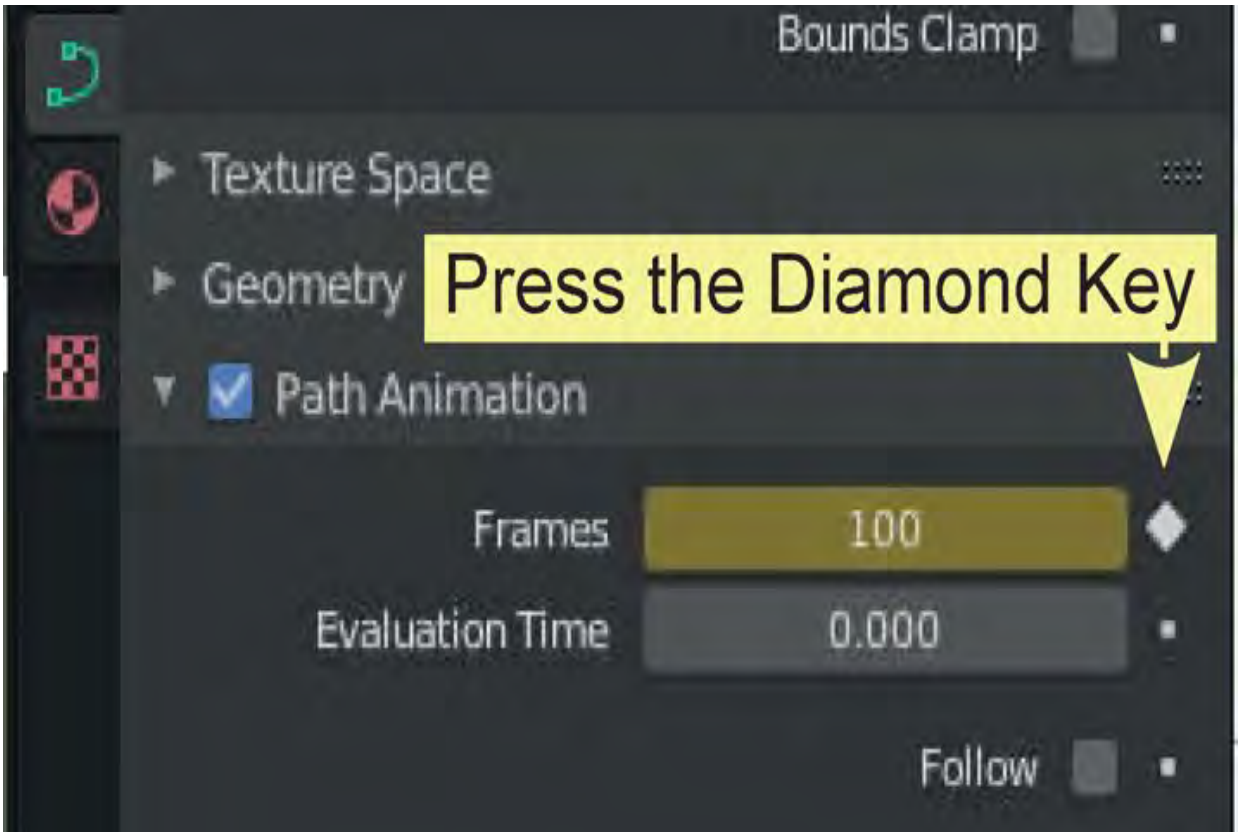


Figure 20.77

Figure 20.78

To have the Cube traverse the Path select the Cube and in the **Follow Path Constraint** click **Animate Path.**

Pressing the Play button in the Timeline shows the Cube following the Path between Frame 1 and Frame 100. At Frame 100 the Cube stops while the Animation continues on to Frame 250 then repeats.

At this point , with the Curve Path selected in the 3D View Editor, the Graph Editor shows two Graph lines; the Evaluation Time (green) and Path Length (red) (Figure 20.80). You will have to zoom and scale the graph Editor to bring the graphs into view.

The red Path Length displays as a straight horizontal line since the motion of the Cube along the Path is constant.

With the Mouse Cursor in the Graph Editor press the N key to open the Properties panel (Figure 20.79) and note, under **Active Keyframes** that the **Interpolation Type** is **Linear.**

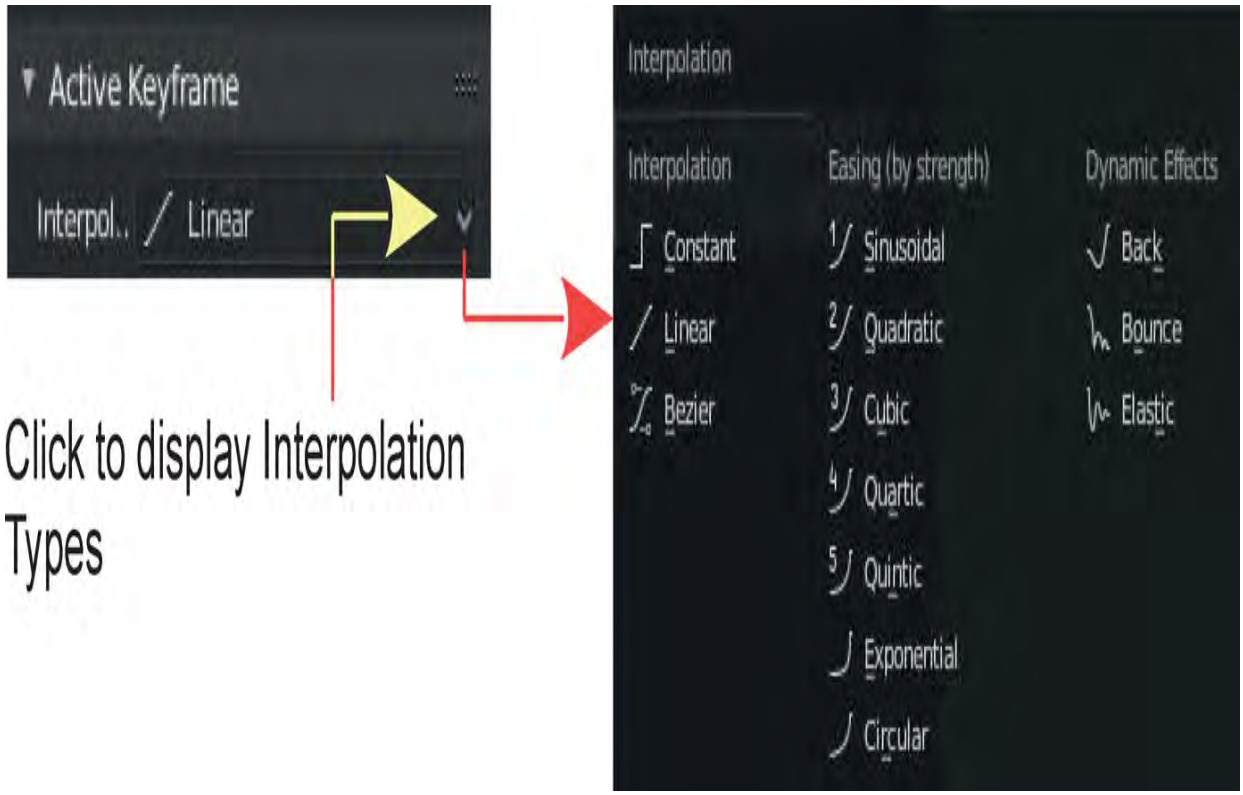


Figure 20.79

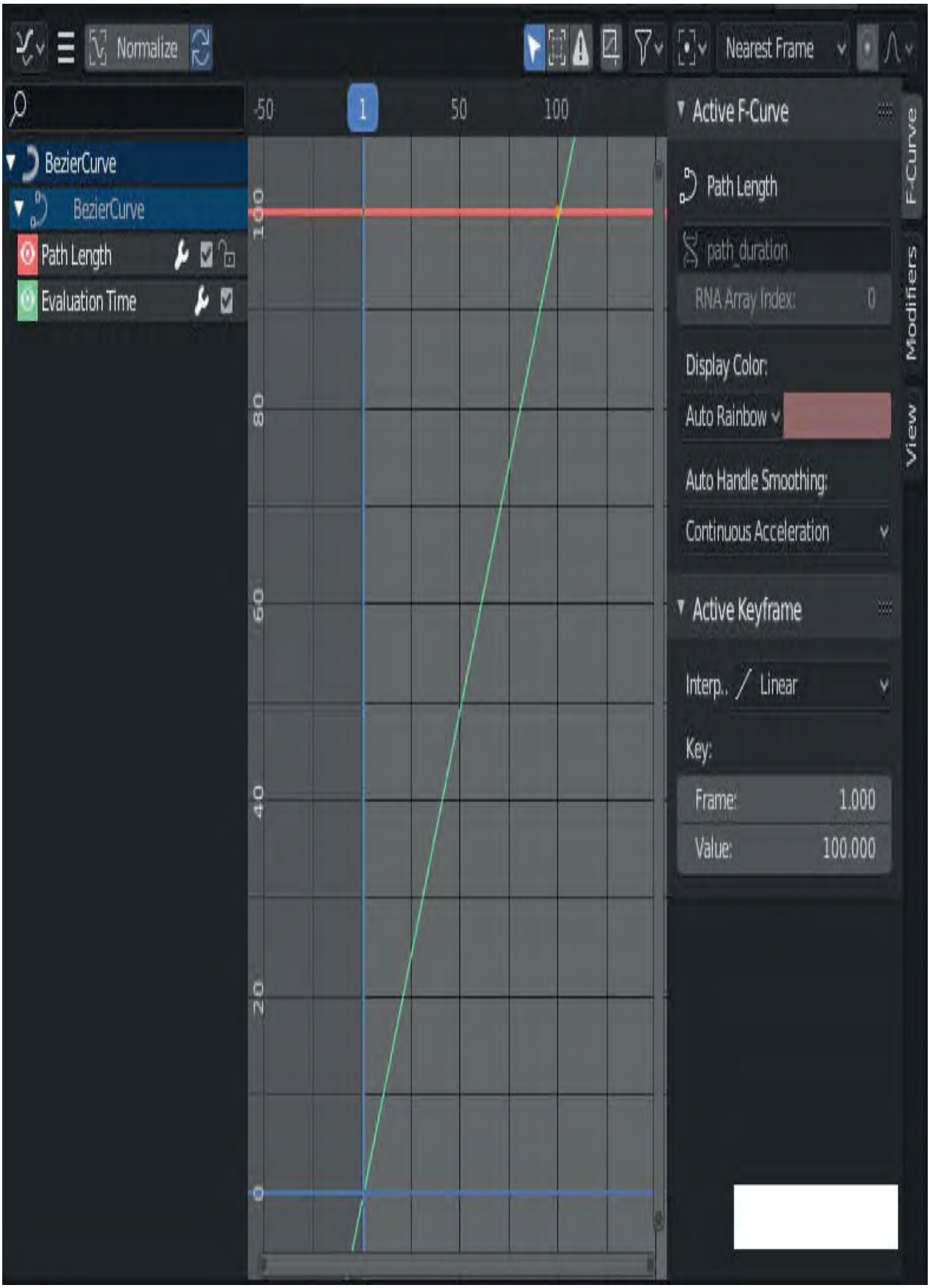

Normalize
Nearest Frame
-50
1
50
100
BezierCurve
BezierCurve
Path Length
Evaluation Time
100
80
60
40
20
0
Active F-Curve
Path Length
path_duration
RNA Array Index: 0
Display Color:
Auto Rainbow
Auto Handle Smoothing:
Continuous Acceleration
Active Keyframe
Interp.. Linear
Key:
Frame: 1.000
Value: 100.000
F-Curve
Modifiers
View

Figure 20.80

Change the Interpolation to Type: Bezier. Press Ctrl and RMB click on the Path Length Graph (red line) to add a control handle then drag down (Figure 20.81).

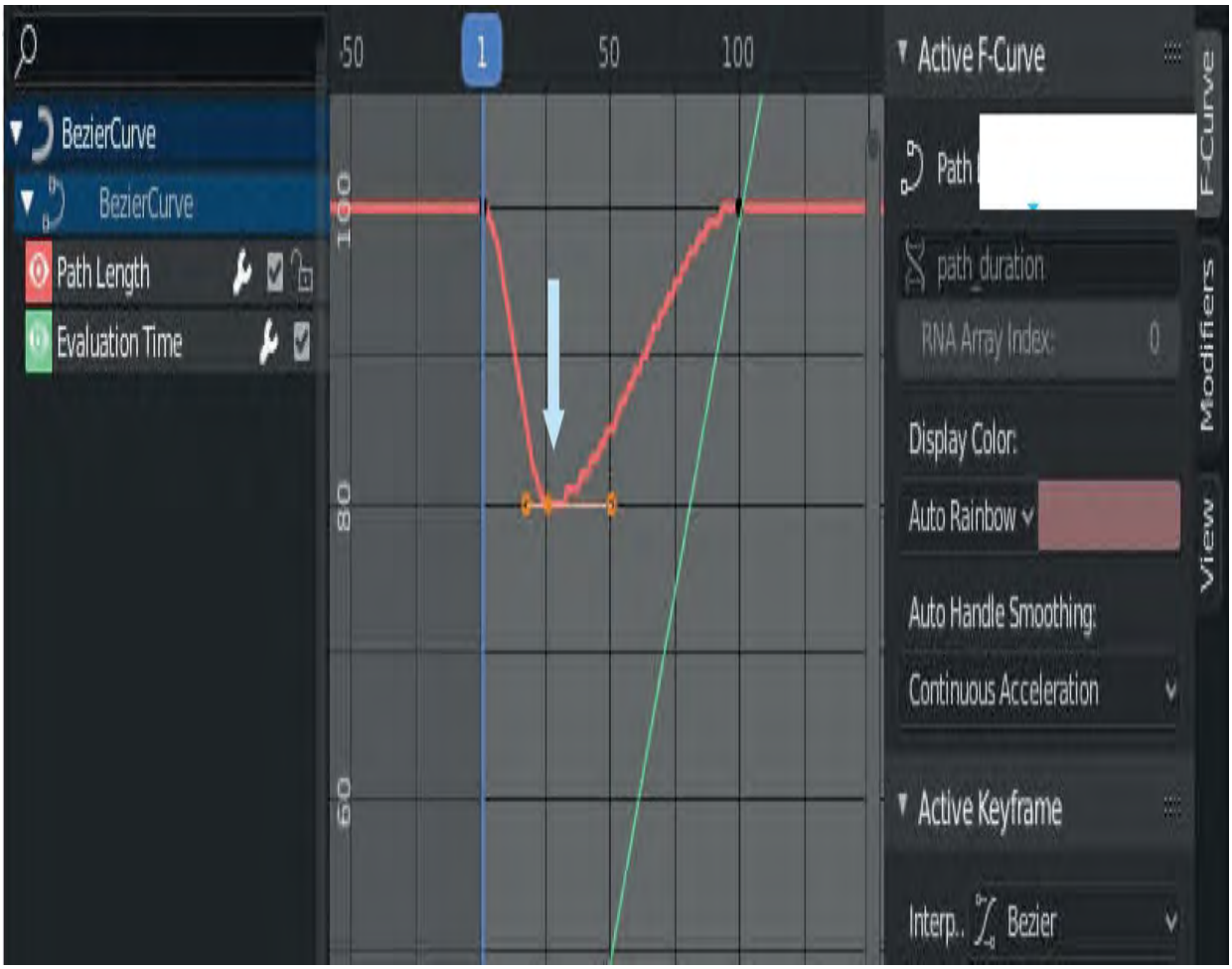


Figure 20.81

Playing the animation show a subtle variation in speed as the Cube moves along the Path, most noticeably, a deceleration as it approaches the end. Adding another control handle and arranging as shown in Figure 20.82 produces a distinct pause in motion.

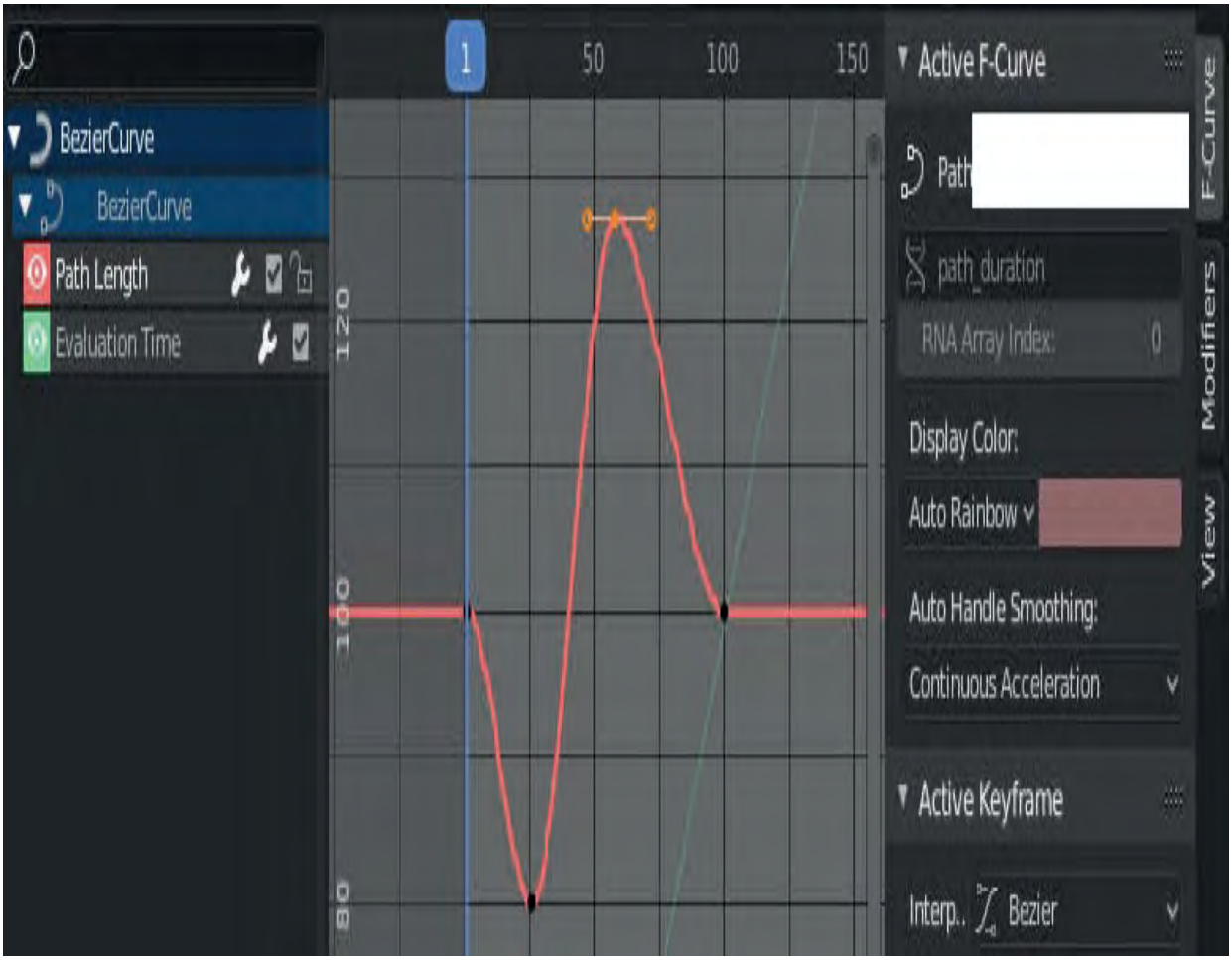


Figure 20.82

If control handles are positioned as shown in Figure 20.83 the Cube will move along the Path until it reaches Frame 85 in the animation then stop. You will observe that this is the point where the **Path Length Graph** intersects the **Evaluation Time Graph** which has become the end of the animation.

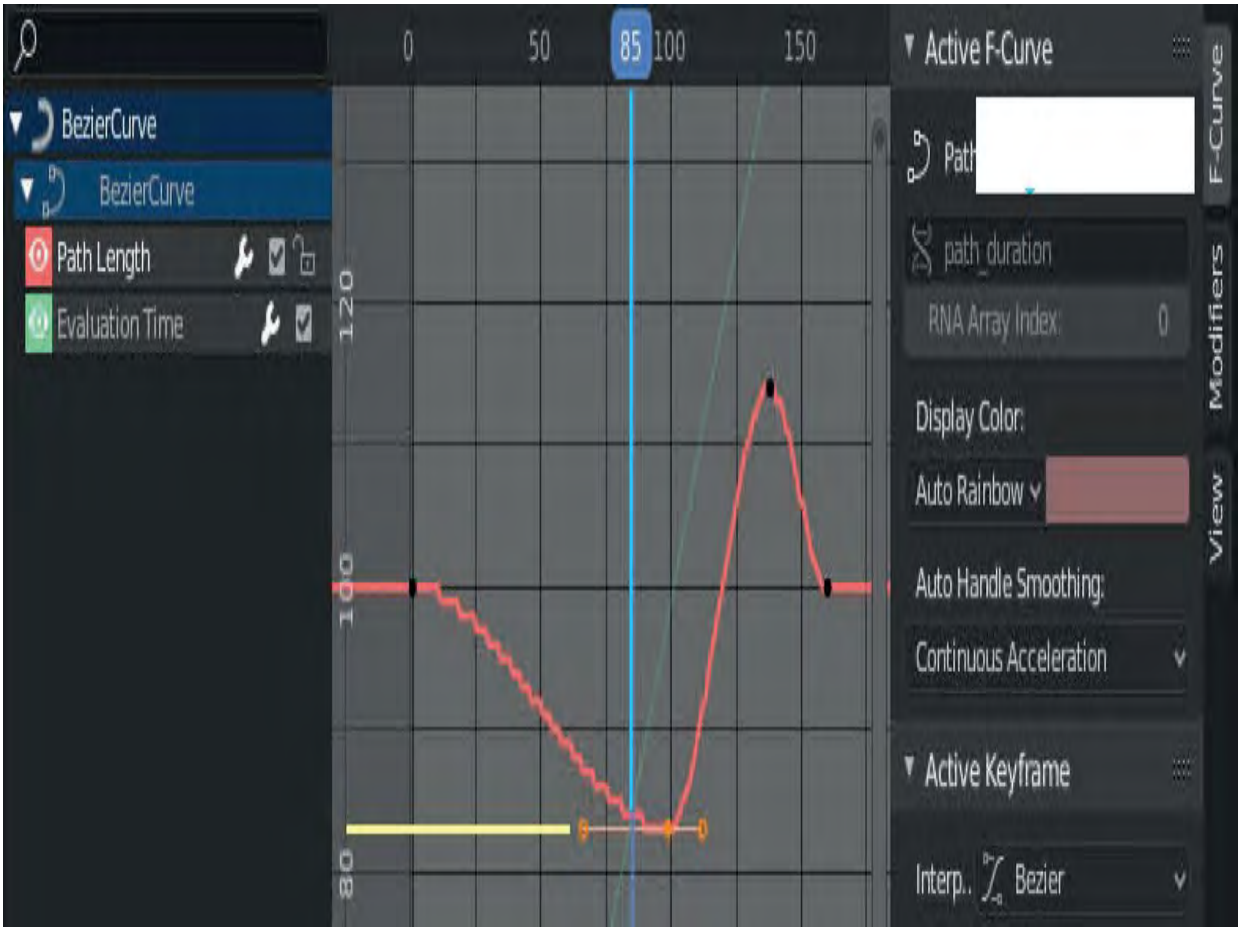


Figure 20.83

Observe that in the Properties Editor Object data buttons for the path Frames = Evaluation Time (Figure 20.84).

Figure 20.84

The foregoing has been an **introduction only** to Path Animation. With the examples shown you will be more easily placed to research and experiment to discover what can be achieved.

20.31 Displacement Sound Animations

A sound file (Music) can be used to affect the movement of vertices producing an interesting display effect.

For a demonstration have an image file (texture) such as that shown in Figure 20.85 and a sound file (music) saved on your hard drive. The Image Texture used in this demonstration is named Art-Fibers.jpg while the sound file is named Flex_Vector_-_Born_Ready.mp3.

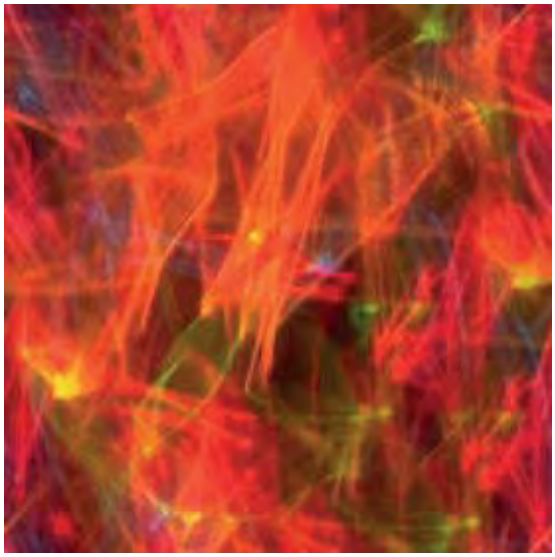


Figure 20.85

Flex Vector is a Hip Hop file with a distinct base beat. You may use any sound file (MP3).

To demonstrate, set up a Sphere object (egg shaped — any shape with a reasonable number of Vertices) (Figure 20.86).

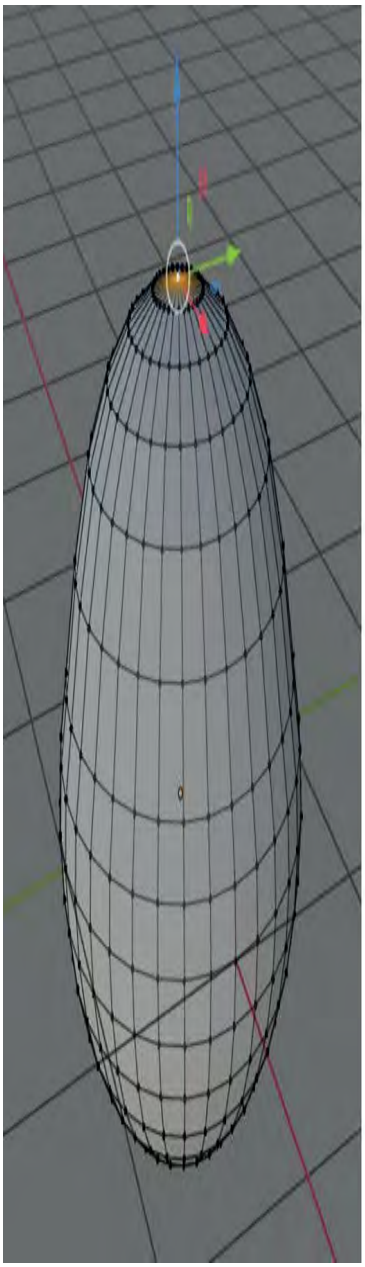


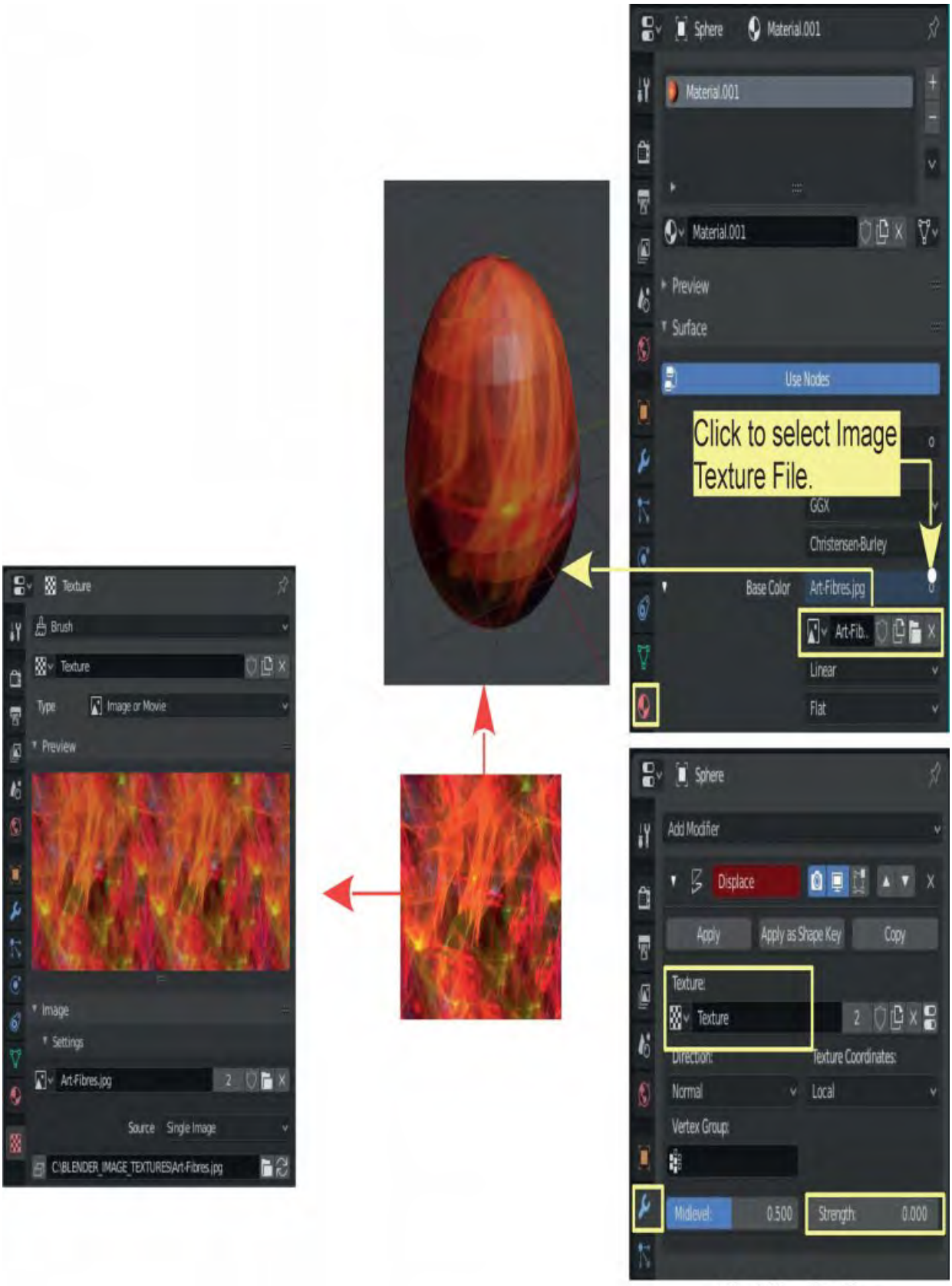
Figure 20.86

3D View Editor Header

UV Sphere with a single Vertex selected and translated up on the Z Axis. **Proportional Editing** has been activated with **Spherical Falloff.**

With the Egg selected in the 3D View Editor, in **Object Mode** go to the **Properties Editor, Material buttons** and **Add a Material. Have Use Nodes active.**

Click the button at the RHS of the Base Color bar and select Image Texture. Navigate to the Texture Image file and click Open Image.



Modifier Buttons

Figure 20.87

Change the 3D View Editor to Rendered Viewport Shading to see the Image applied as a Material to the Egg.

With the Egg selected in the 3D View Editor go to the Properties Editor, Texture buttons and Add a Texture, selecting the same Image as before.

The Texture will be named Texture.

In the Properties Editor, Modifier buttons Add a **Displace** Modifier to the Egg selecting the Texture named **Texture.**

With the Strength value in the Modifier: 1.000 the Egg in the 3D Viewport Editor is deformed by the Texture. Set the Strength value to 0.000.

Figure 20.88

Divide the 3D Viewport Editor horizontally and make the lower part the **Video Sequence Editor.** The Timeline Editor remains at the bottom of the Screen.

With the Egg selected and the Timeline Editor Cursor at Frame 1 in the Timeline, in the Displace Modifier panel RMB click on the Strength value and select Insert Keyframe.

The Strength value slider turns yellow and a Keyframe is entered in the Timeline at frame 1.

In the Video sequence Editor Header click Add — Sound and navigate to the sound file then click Add Sound at the upper RHS, The file is entered in Channel 1 in the Video Sequence Editor.

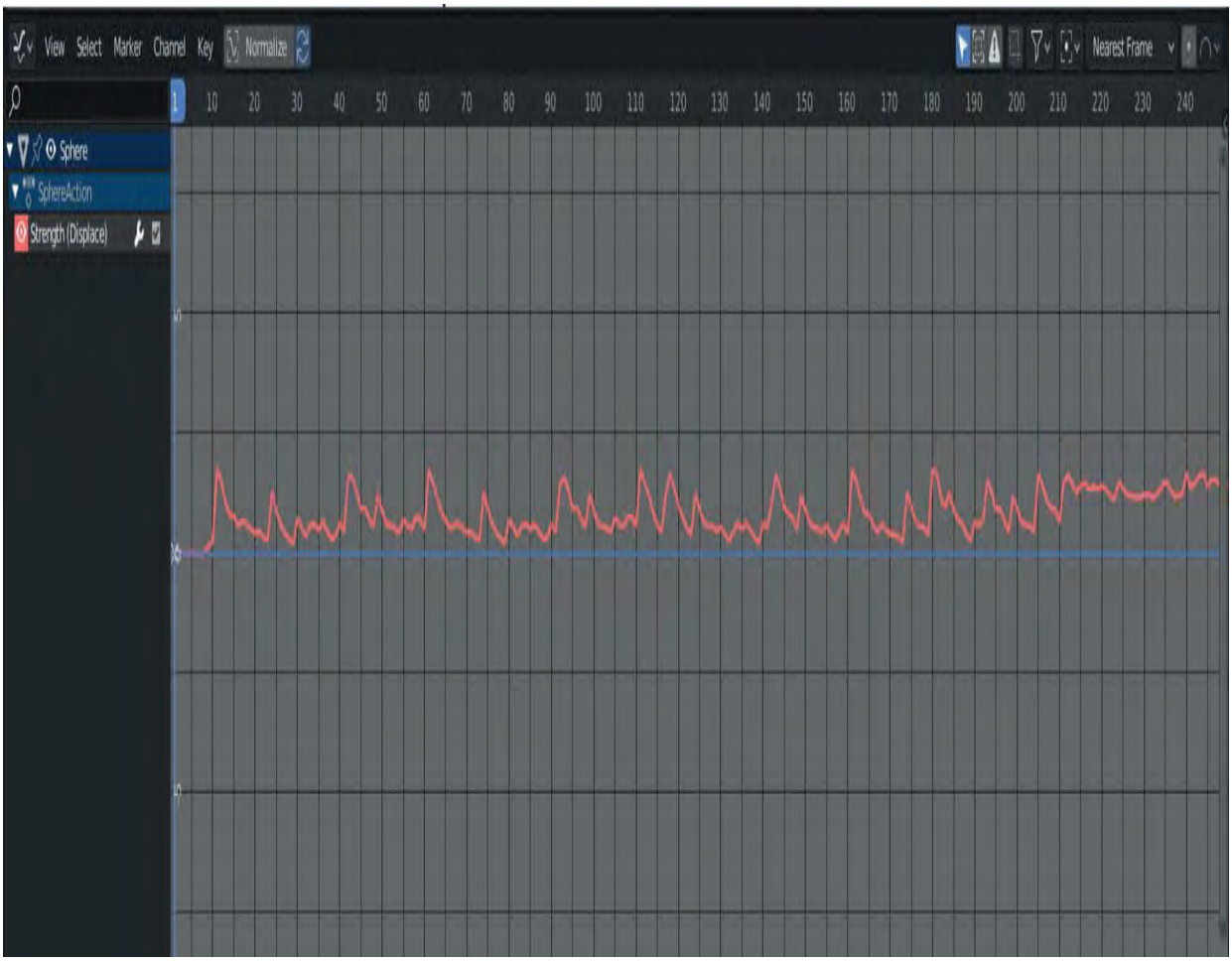
Sequencer
View
Select
Marker
Add
Frame
Strip
Backdrop
Refresh Sequencer
Scene
Clip
Mask
Movie
Image
Sound
Generate
Effect Strip
MP3 Sound File
Flex_Vector_-_Bom_Ready.mp3: //..\..\Users\John\Downloads\Flex_Vector_-_Bom_Ready.mp3 | 6602
00:01
00:02
00:03
00:04
00:05
00:06
00:07
00:08
00:09
00:10

Figure 20.89

Change the Video Sequence Editor to the **Graph Editor** and in the Header click **Key** and select **Bake Sound to F-Curve.** Navigate to and select the Sound File then click **Bake Sound to F-Curve** in upper RHS of the panel. **Note:** A Keyframe must be entered before this instruction.

Figure 20.90

A Sound Curve is drawn in the Graph Editor.

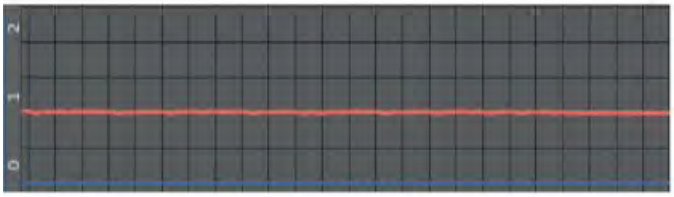


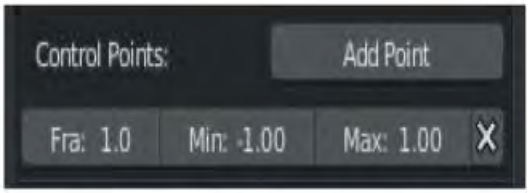
Playing the animation sees the Egg pulsate in the 3D Viewport Editor to the beat of the Sound (have the speakers turned on).

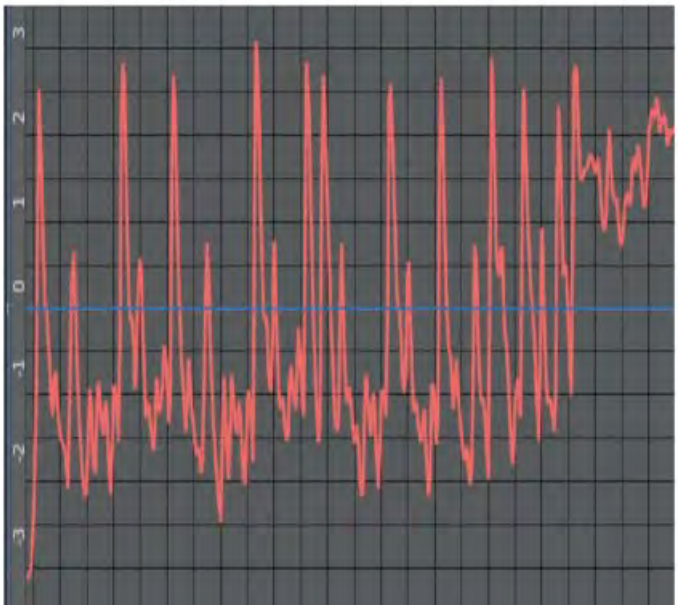
This result may be what is required but if not you can modify the Curve in the Graph Editor. With the Mouse Cursor in the Graph Editor press the N Key to display a Properties Panel and select the Modifiers tab.

Click Add Modifier and select Envelope. Click Add Point to add control points. Adjusting the Control Point values alters the Sound Curve in the Graph Editor which alters the way in which the sound affects the animation.

Modifiers
Add Modifier
Envelope
Envelope:
Reference Value: 0.000
: -1.000
M: 1.000
Control Poi..
Add Point
Restrict Frame Range
Use Influence
F-Curve
Modifiers
View


2
1
0


Control Points:
Add Point
Fra: 1.0
Min: -1.00
Max: 1.00


3
2
1
0
-1
-2
-3

Control Points:
Add Point
F: 100.0
Min: -6.60
Max: 0.40

Figure 20.92

20.32 Sound Effect and Cast Modifier

Displacement Sound Animation in the previous section used a sound file to affect the **Strength value** of a **Displacement modifier.** A sound file was then **Baked** to an **F-Curve** and modified. This was to control the displacement of the Object's surface.

You may combine a sound file **F-Curve** and a **Cast modifier** with an **Empty** control object to influence an animation of the Object's surface deformation.

To demonstrate set up a Plane object with a **Cast Modifier** and an **Empty** object (Figure 20.93).

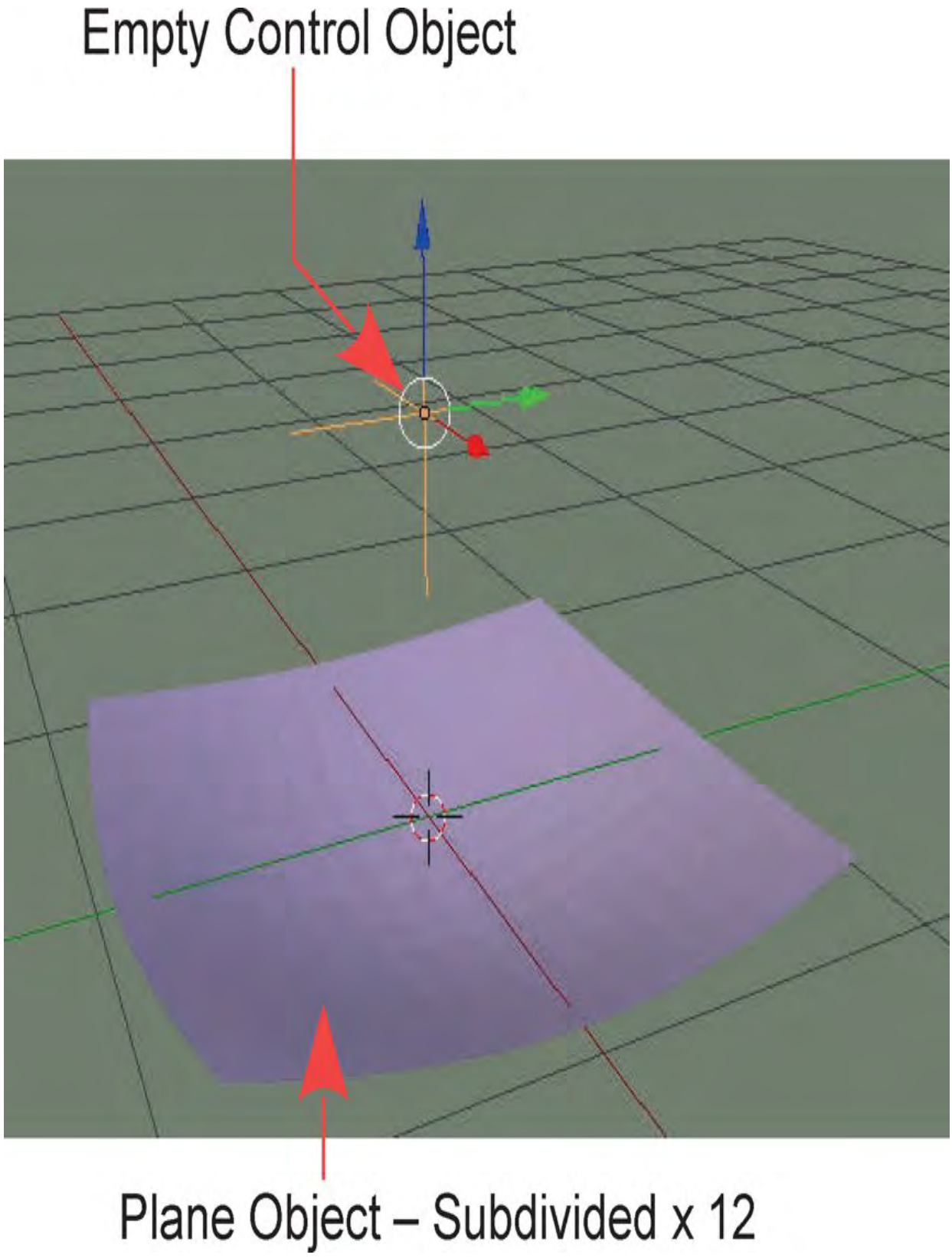


Figure 20.93

Set up the 3U View editor

Properties Editor

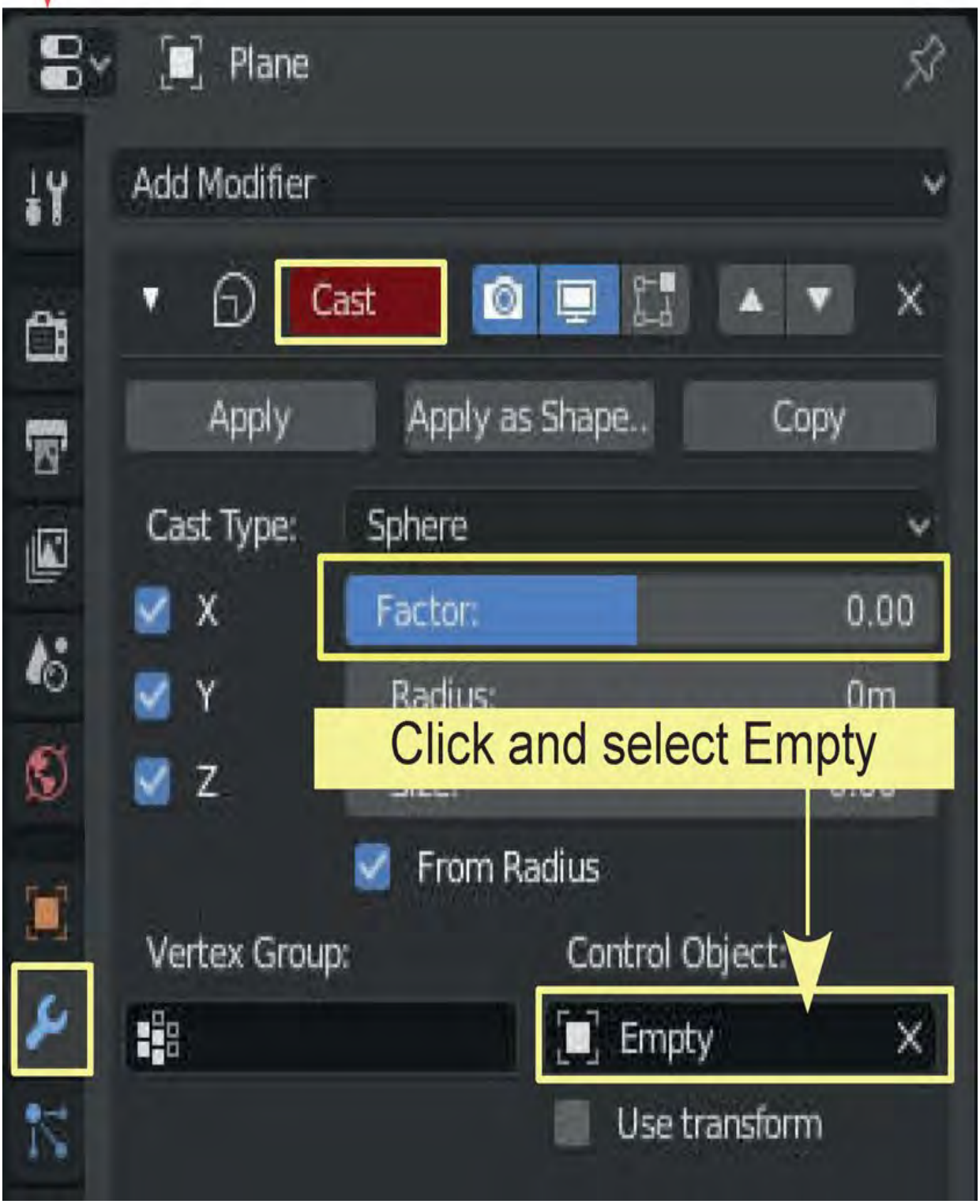


Figure 20.94

Add a Sound File to the **Video Sequence Editor.**

Create an F-Curve for the **Factor value** of the Cast modifier by selecting the Plane, then settingthe **Factor value** in the **Cast modifier to 0.000.** In the **Timeline Editor** position the cursor at Frame 1. Right click on the **Factor value slider** in the Cast Modifier and select **Insert Keyframe.**

Declare the sound file to affect the Factor value by dividing the 3D Viewport Editor in two and changing one half to the **Graph Editor.** In the Header click on **Key** and select **Bake Sound to F Curves.** Navigate to the sound file, select the file and click Bake Sound to F-Curves in upper **RH** corner of the Editor. The bake can take a while. The sound file **F-Curve** is inserted in the **Graph Editor.**

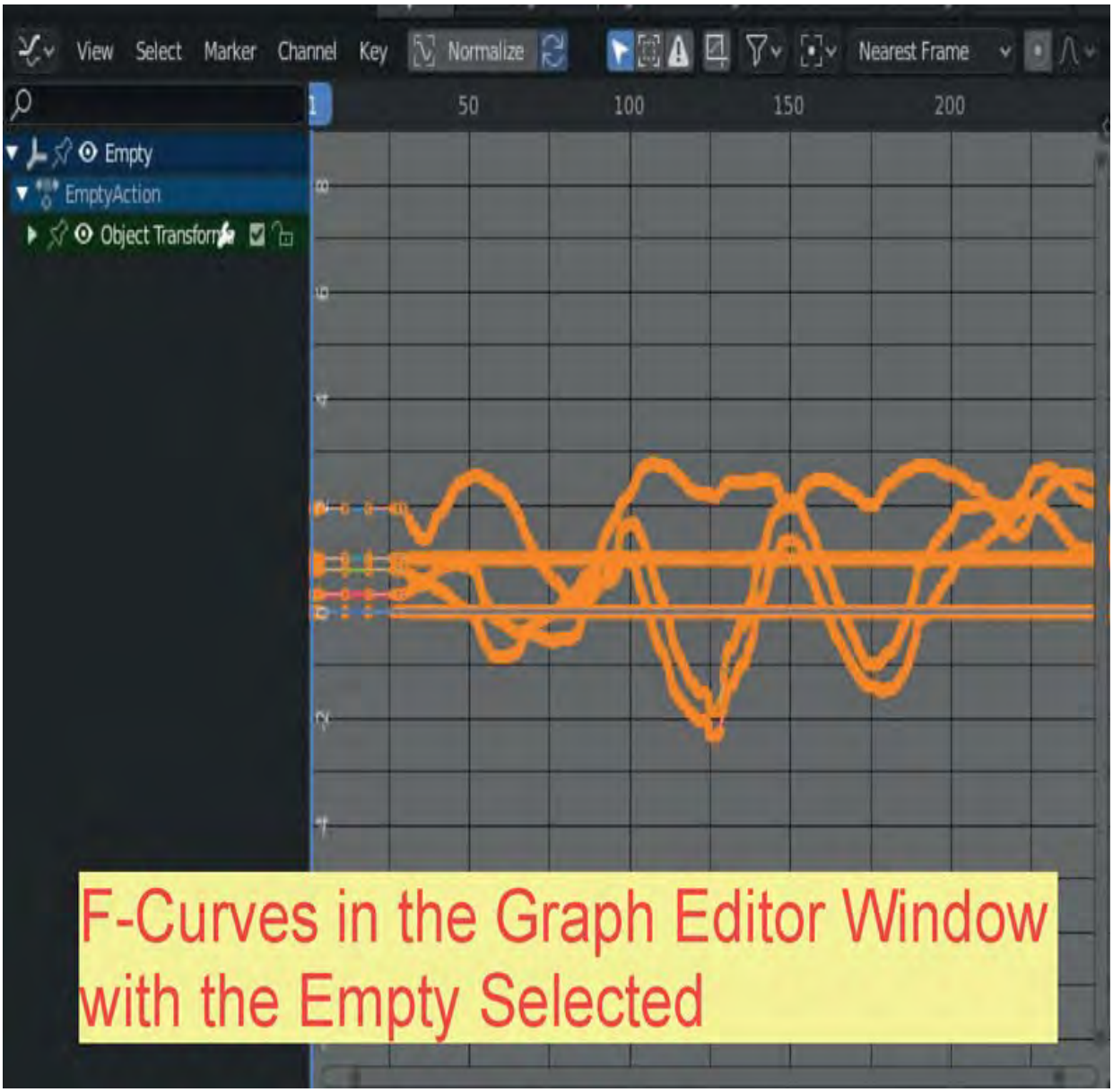


Figure 20.95

Automatically Insert Keyframes

Turn on Auto Keyframing in the Timeline Editor Header. With the **Empty Object** selected in the 3D Viewport Editor play the animation and at the same time press **G Key** in the **3D Viewport Editor** and move the Empty object about. Keyframes are added to the Timeline. Stop the animation, reinstate the sound file in the Video Sequence Editor and replay to see the effect.

Keyframes Added in the Timeline Window

Figure 20.96

Keyframes Added in the Timeline Window

Figure 20.97

Sound file in the Video Sequence Editor

When the Animation is played the Mesh Deforms to the beat in the 3D Viewport Editor.

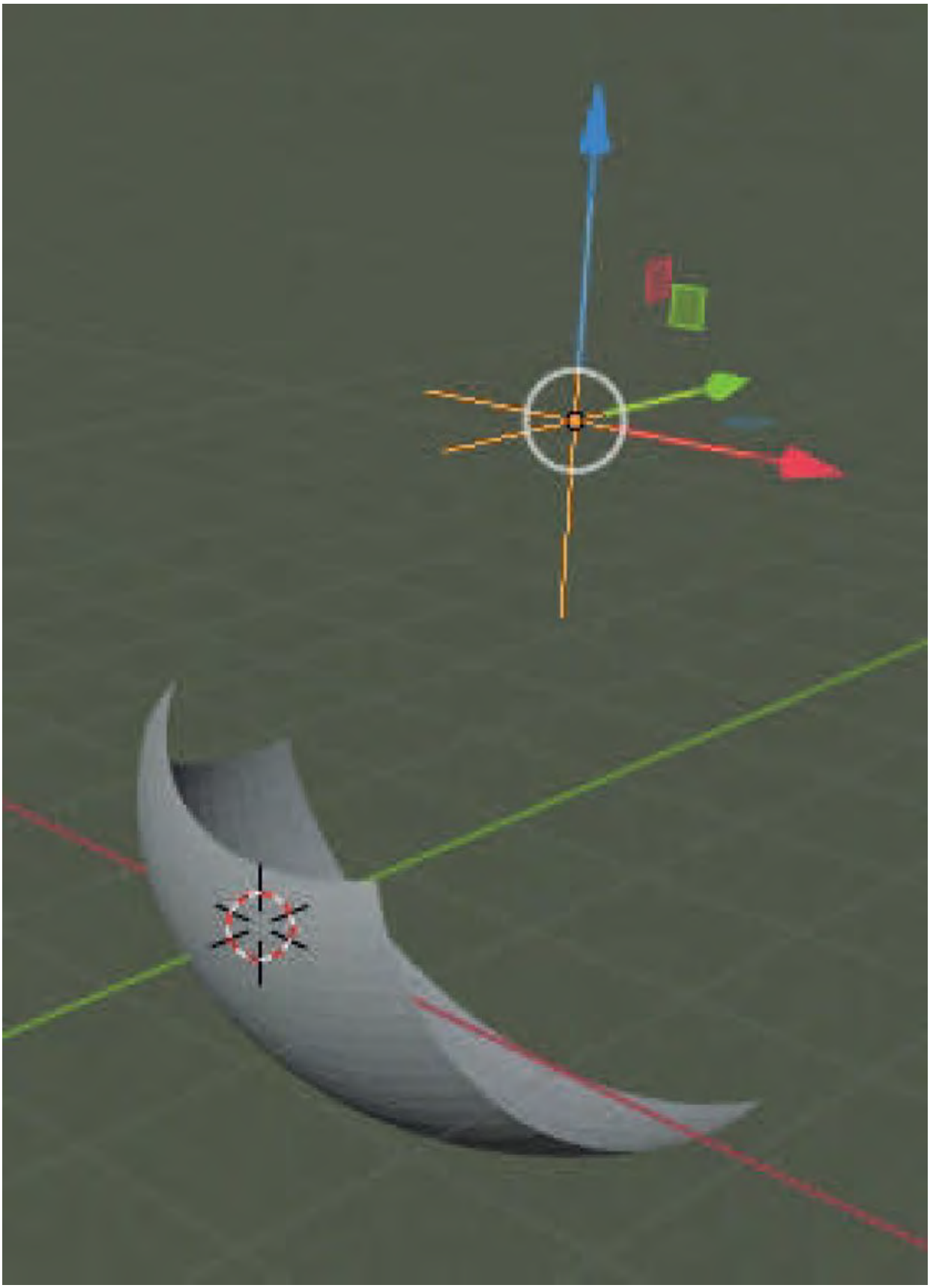


Figure 20.98

20.33 The Animation Workspace

The **Animation Workspace** is a Screen arrangement comprising, a 3D Viewport Editor in Camera View, a 3D Viewport Editor in User Perspective View, an Outliner Editor, a Properties Editor, the Dope Sheet Editor and the Timeline Editor.

All of the operations and examples described in the sections of this chapter are applicable to Editors making up the Animation Workspace.

Figure 20.99

The arrangement of Editor panels in the Workspace may or may not suit your particular way of working, therefore, the preceding explanation of working different applications in different Editors will allow you to configure your personalised Workspace.

The Animation Workspace

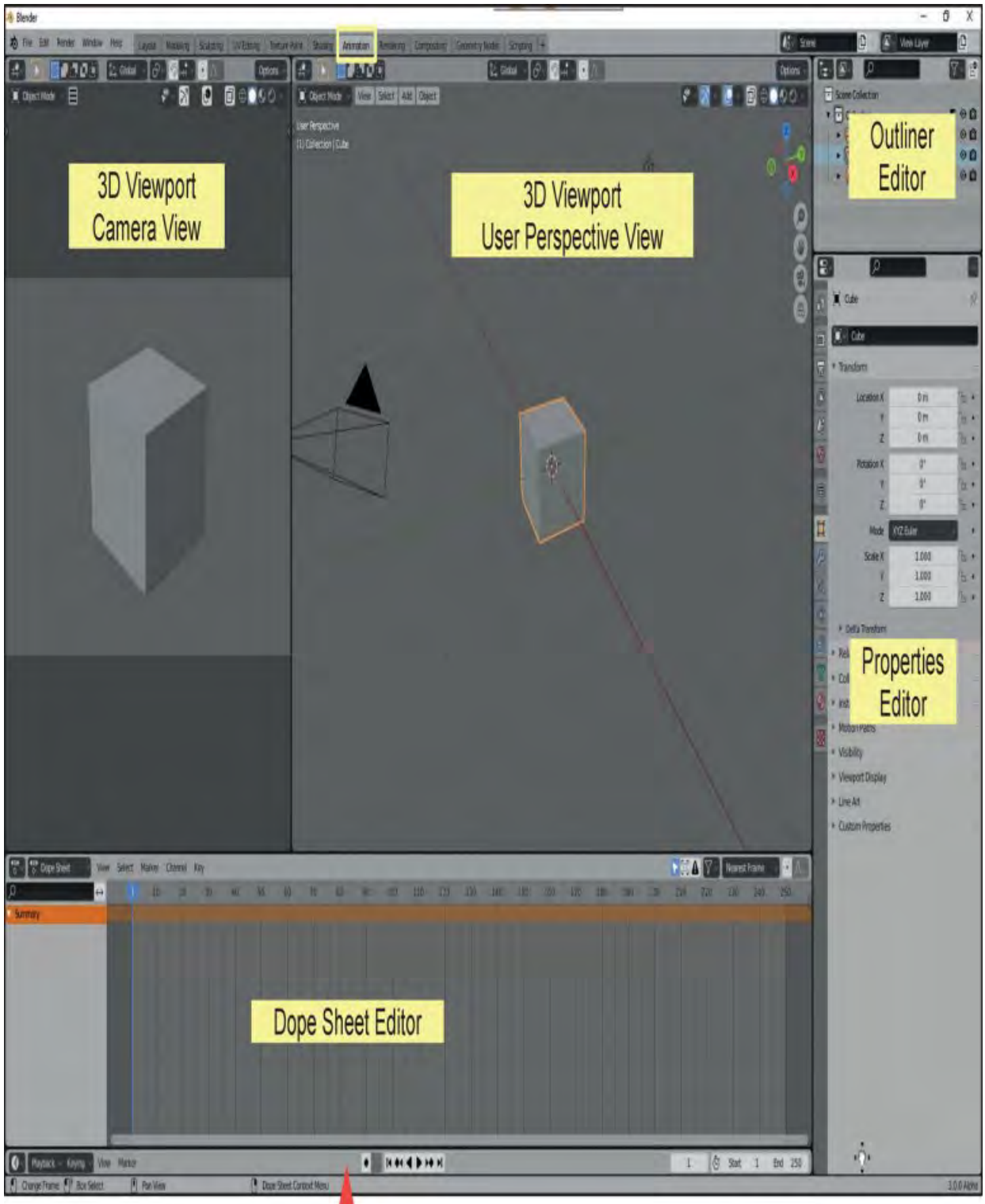

3D Viewport
Camera View
3D Viewport
User Perspective View
Outliner
Editor
Properties
Editor
Dope Sheet Editor
Timeline Editor – Header onl Displayed
(Drag the Border Up to see the full Editor

Figure 20.100

Note: The Animation Workspace is shown with **Blender Light Theme** Active (Chapter 2-2.21).

21 Constraints

21.1 Introduction to Constraints
21.2 Track to Constraint
21.3 Constraint Stack
21.4 Transform Constraint List
21.5 The Transform Constraint
21.6 Tracking Constraints
21.7 Relationship Constraints
21.8 The Action Constraint
21.9 The Shrinkwrap Constraint
21.10 Follow Path – Extrude/Bevel

Constraints control an Object's properties such as its Location, Rotation and Scale by Targeting the Object to a secondary Object or connecting Objects in a Scene together, such that they act as a single entity while maintaining individual characteristics. Another way to define Constraints is to say they define relationships between Objects.

For example, the **Track To Constraint** applied to a **Camera** Object, with a **Target** set as another Object in the Scene, causes the Camera to always point to the second Object no matter where it moves.

Another example is a **Child Of Constraint** which when applied to one Object (the child) with a **Target** set as a second Object (the parent) causes the child to follow the parent. Using a Child Of Constraint creates what is termed a **Child / Parent Relationship.** This has a particular application in animating characters.

Constraints are applied to Objects in the **Properties Editor, Object Constraint Properties** by clicking on **Add Object Constraint** then selecting a Constraint from the menu that displays.

Constraints in Blender are listed in four categories as shown in Figure 21.2 on the following page.

In this chapter, Constraints are briefly defined and several examples provided which will allow you to understand their application.

21.1 Introduction to Constraints

Constraints are accessed in the **Properties Editor, Object Constraint Properties** (Figure 21.1).

Clicking **Add Object Constraint** displays the Constraint selection menu listing Constraints in four categories (Figure 21.2).

Figure 21.1
Figure 21.2

21.2 Track to Constraint

The Track To Constraint provides an introduction demonstrating what a Constraint is, in practical terms and how Constraints are used.

In the default Blender Scene a Camera Object is directed towards the Cube Object such that it captures the Cube in Camera View (Num Pad 0). When the Cube is animated to move across the Screen the Cube can move in and out of Camera View. If you want the Cube to remain in view no matter where the Cube is in the Scene, you track the Camera to the Cube by employing the **Track To Constraint** (add the Constraint to the Camera).

Note: The default camera has been rotated and locked in position to point towards the center of the Scene (default position of the Cube). The rotation of the default Camera has to be unlocked to use the Track To Constraint. This only applies to the default Camera. A new camera entered in the Scene is not locked.

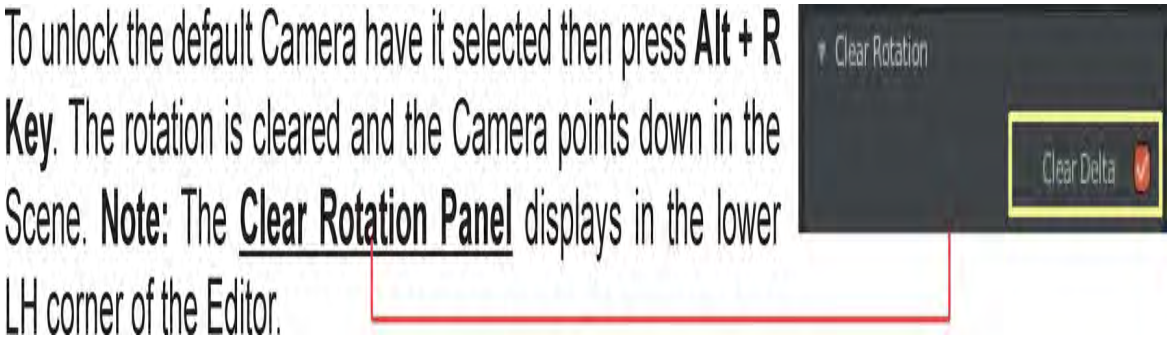


Figure 21.3

If **Delta Transform Rotation** values have been entered in the **Properties Editor, Object Properties, Delta Transform Tab,** check **Clear Delta** in the Clear Rotation Panel.

With the Camera selected click **Add Object Constraint** (Figure 21.1) and select **Track To** in the menu (Figure 21.2).

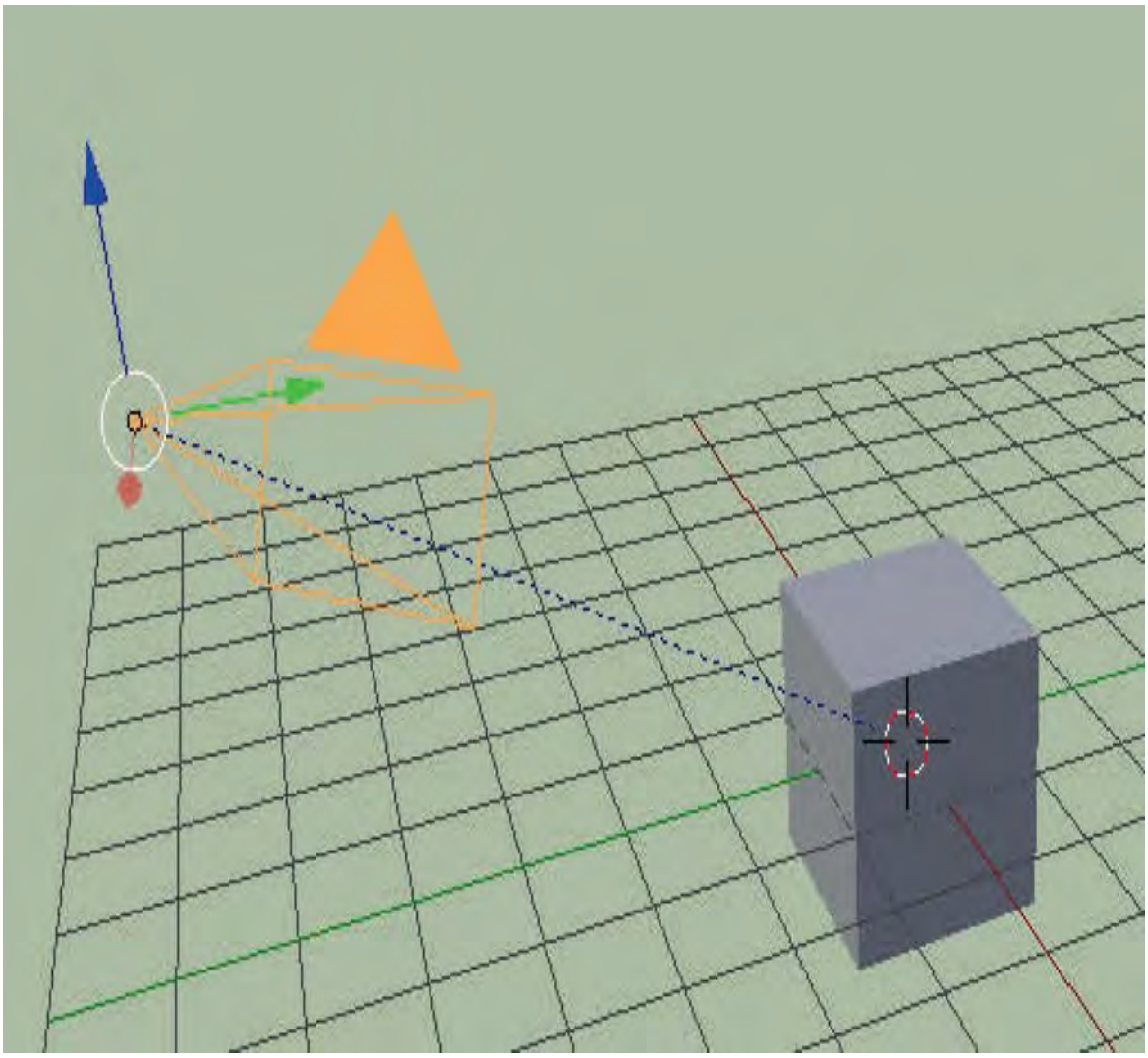


Figure 21.4

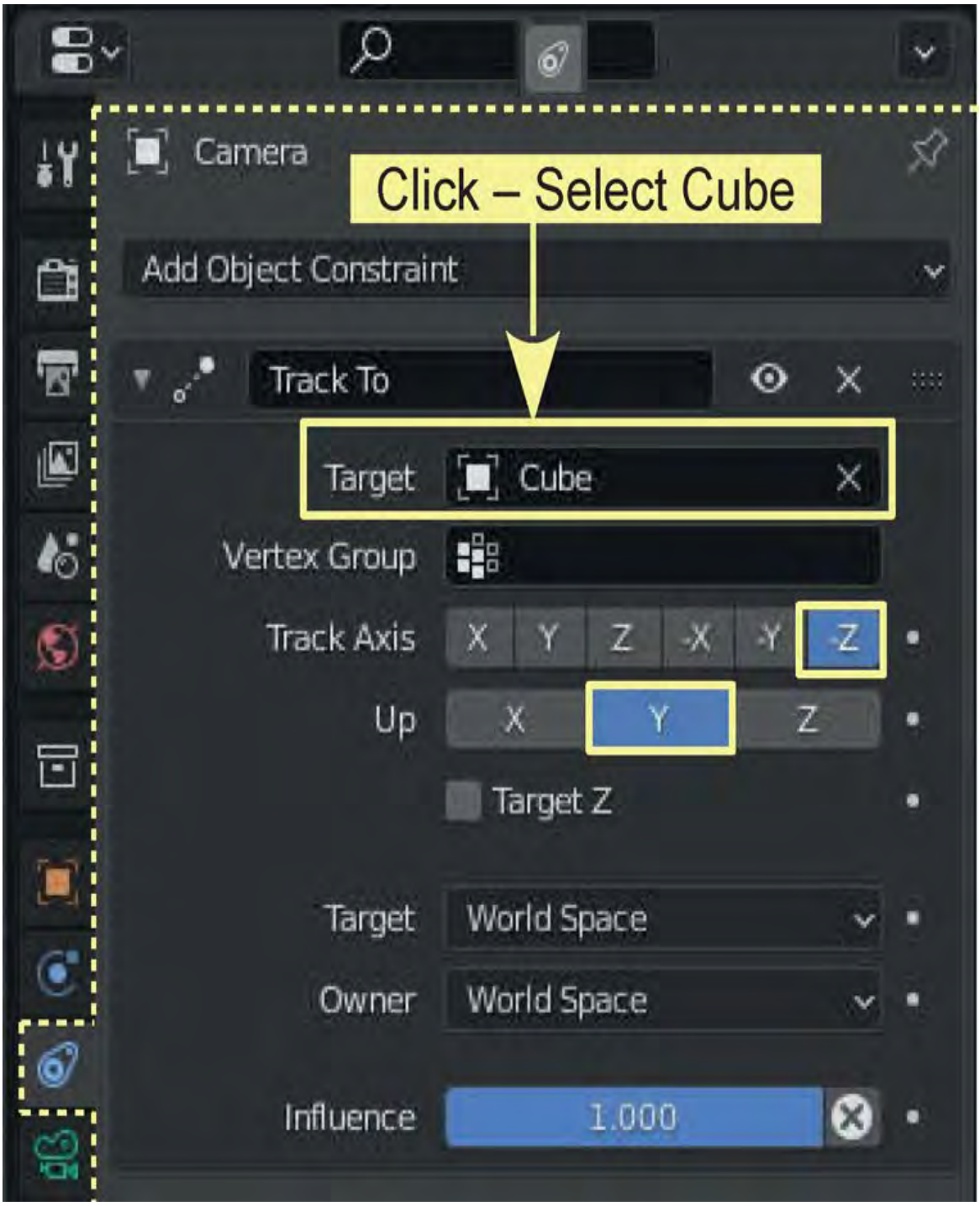


Figure 21.5

In the **Track To** panel click on **Target** and select the **Target Object** (Cube) in the menu (Figure 21.5).

Note: On entering the **Target** (Cube) the Camera swings around pointing to the Cube. There is a broken line connecting the Camera to the Cube indicating that a Constraint is applied. Note the **Track Axis** and **Up** directions in the Constraint Panel (Track Axis:-Z and Up: Y) (Figure 21.5). With the Cube animated to move in the Scene the Camera always points to the Cube.

Constraints are associated with an Object by selecting the Object in the 3D Viewport Editor then clicking on **Add Object Constraint** in the **Properties Editor, Object Constraints Properties** and selecting the Constraint from the menu that displays (Figure 21.2).

21.3 Constraint Stack

It should be noted, in some cases, it is appropriate to apply more than one Constraint to an Object. When this is done the Constraints are placed in a stack in order of priority. The priority can be changed by moving a constraint up or down in the stack (Figure 21.6).

In Figure 21.6 a **Follow Path** and a **Track To** Constraint are applied to the same Object. The **Track To** takes precedence over the **Follow Path**. To reverse the precedence click, hold and drag the **dimple** button up or down.

Figure 21.6

In many cases, when using Constraints, there are control values to be inserted in the Constraint Panel to regulate the functions. The following pages in this chapter contain a brief description of Constraint functions. Most Constraints are self explanatory, therefore a detailed explanation will only be given for a few common Constraints, or where it is not self evident.

21.4 Transform Constraints List

- **Copy Location.** Forces the Object with the constraint added to take up the location of the Target Object.
- **Copy Rotation.** Forces the Object with the constraint added to copy the rotation of the Target Object. When the target rotates, the Object rotates.
- **Copy Scale.** Forces the Object with the constraint added to copy the scale of the Target Object
- **Copy Transforms.** Similar to the copy location constraint.
- **Limit Distance.** Constrains the Object to remain within a set distance from the Target Object. The distance is a spherical field surrounding the target and the Object is constrained within or outside the spherical field.
- **Limit Location.** Constrains the Object's location between a minimum and maximum distance on a specific axis. The distance is relative to either the world center or a parented Object.
- **Limit Rotation.** Constrains an Object's rotation about a specific axis between limits.
- **Limit Scale.** Constrains the scale of an Object between limits on a specified axis.
- **Maintain Volume.** Constrains the dimensions of a side on a specified axis.
- **Transformation.** See Section 21.5. Note: In 21.5 the Properties Editor is shown using the Blender Light, Preset Theme (see Chapter 2 – 2.21 Themes, Figure 2.42).
- **Transform Cache.**

21.5 The Transformation Constraint

The **Transformation Constraint** allows you to control the Location, Rotation or Scale of one Object or part of an Object by adjusting the Location, Rotation, or Scale of another Object. The Location, Rotation or Scale values in either case can be set to operate within a specific range.

The Object to be controlled is called the **Source** and has the **Constraint** applied to it, while the other Object (the controlling Object), is called the **Target object.**

To demonstrate have a Cube Object and a UV Sphere in the 3D Viewport Editor in Top Orthographic View (Figure 21.7)

The **Cube** will be the **Source Object** controlled by the **Target Sphere. Add a Transformation Constraint to the Cube.**

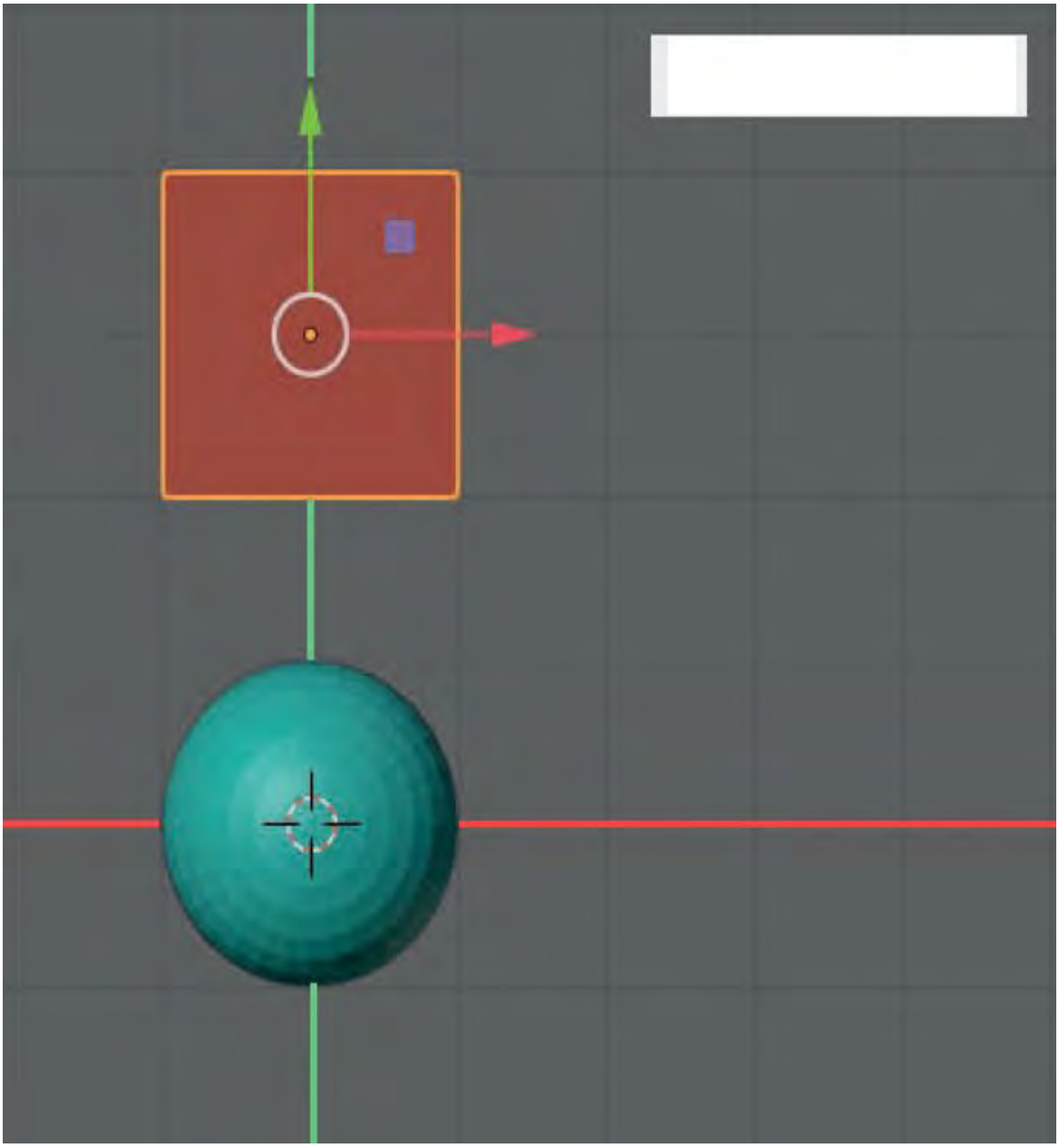
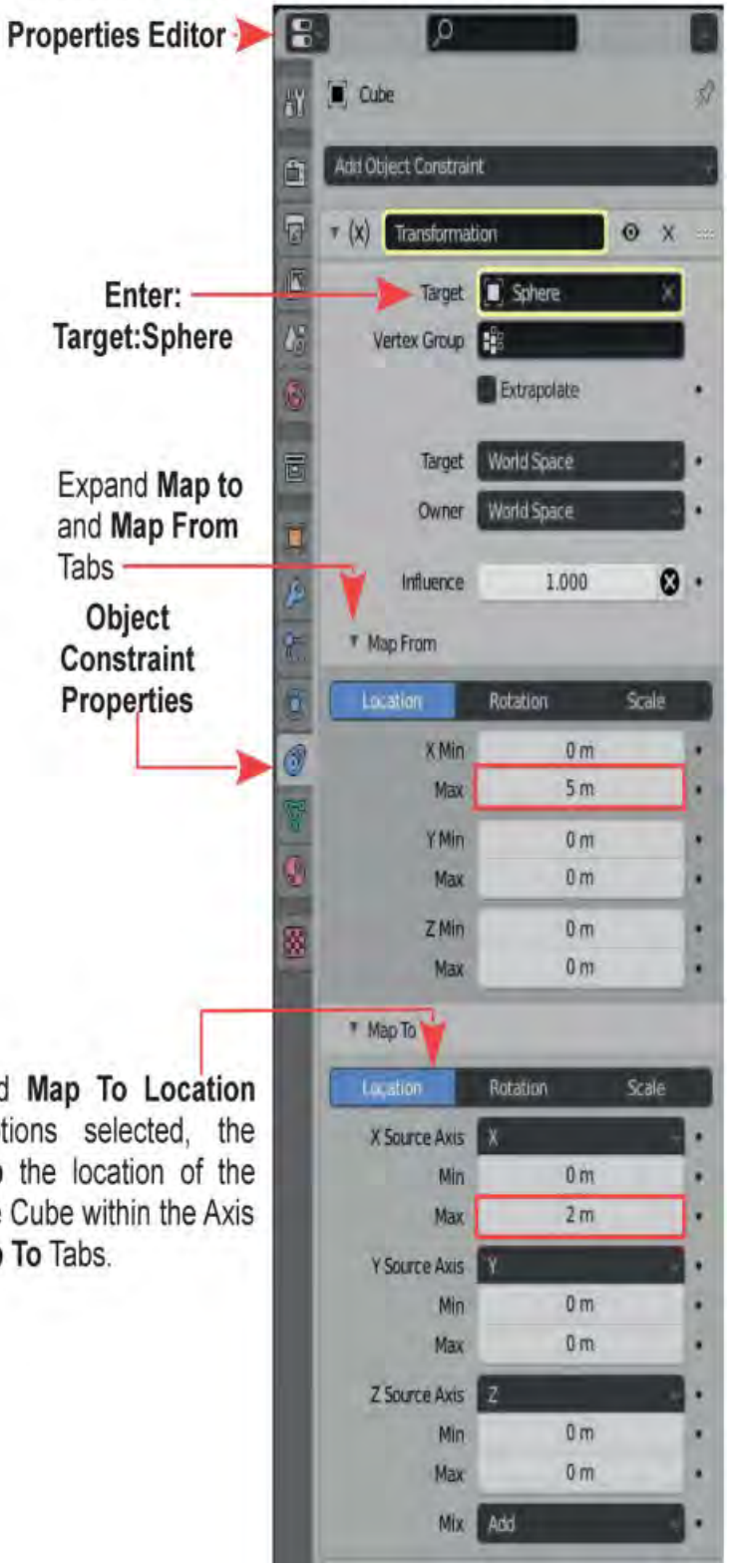


Figure 21.7

To apply the **Constraint,** select the Source (Cube), click **Add Object Constraint** in the **Properties Editor, Object Constraints Properties** and select **Transformation** under the **Transform** heading. Enter **Target: Sphere** (Figure 21.8).

Object and a UV Sphere in the 3D Viewport Editor in Top



Note: The default **Map From** and **Map To Location** options. With the **Location** options selected, the philosophy of operation is to **Map** the location of the Sphere relative to the location of the Cube within the Axis range set in the **Map From** and **Map To** Tabs.

Figure 21.8

In this example, Translating the Target Sphere between Om and 5m along the X Axis of the Scene will affect the Translation (Location) of the Cube between Om and 2m along the X Axis.

You see this effect in practise in Figure 21.9.

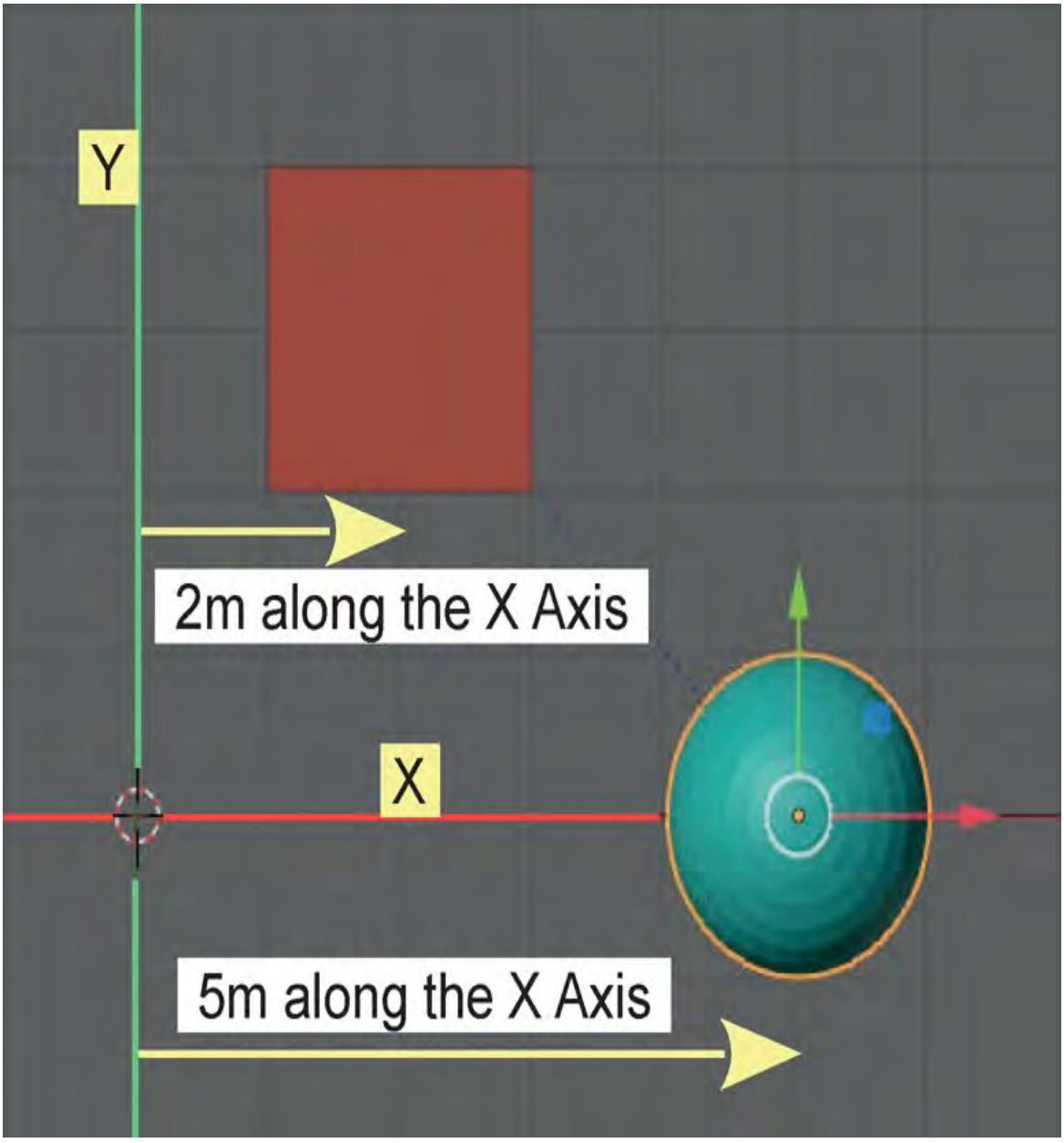


Figure 21.9

As the Sphere is Translated along the X Axis the Sphere is Constrained such that it follows. The Translation only occurs when the Sphere is translated between Orn and 5m on the X Axis and the Cube only moves between Orn and 2m.

By configuring the Map From and Map To values in the Constraint Properties the movement of one Object is used to affect the movement of another.

Consider the following example.

With the initial arrangement as shown in Figure 21.7 and with values revised in the **Map From** and **Map To** Tab as shown in Figure 21.10, when the Target Sphere is Translated between Orn and 5m along the X Axis the Source Cube moves in a negative direction towards the X Axis of the Scene.

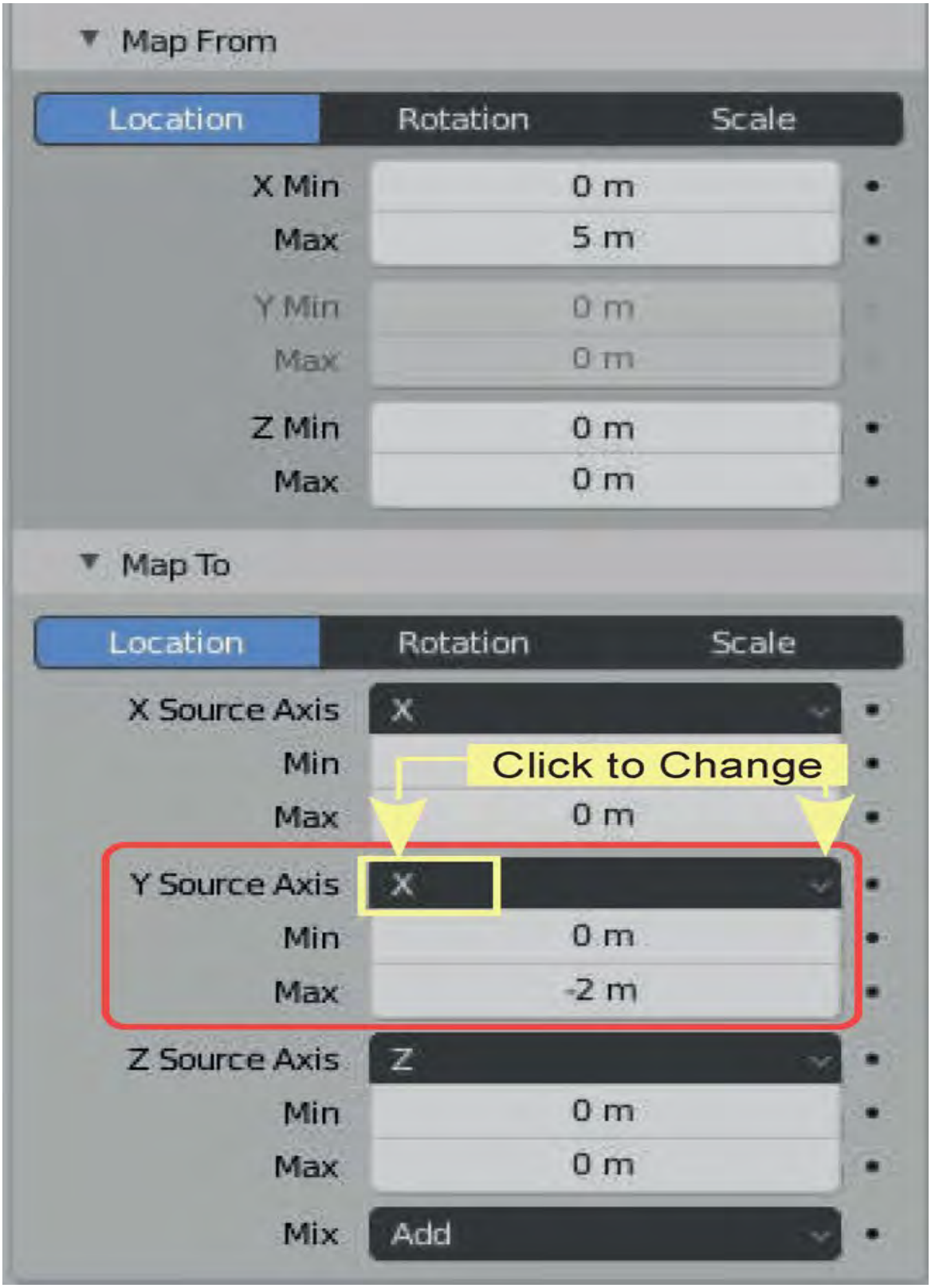

Map From
Location
Rotation
Scale
X Min
0 m
Max
5 m
Y Min
0 m
Max
0 m
Z Min
0 m
Max
0 m
Map To
Location
Rotation
Scale
X Source Axis
X
Min
Click to Change
Max
0 m
Y Source Axis
X
Min
0 m
Max
-2 m
Z Source Axis
Z
Min
0 m
Max
0 m
Mix
Add

Figure 21.10

When the Sphere is Translated the Cube moves towards the X Axis of the Scene.

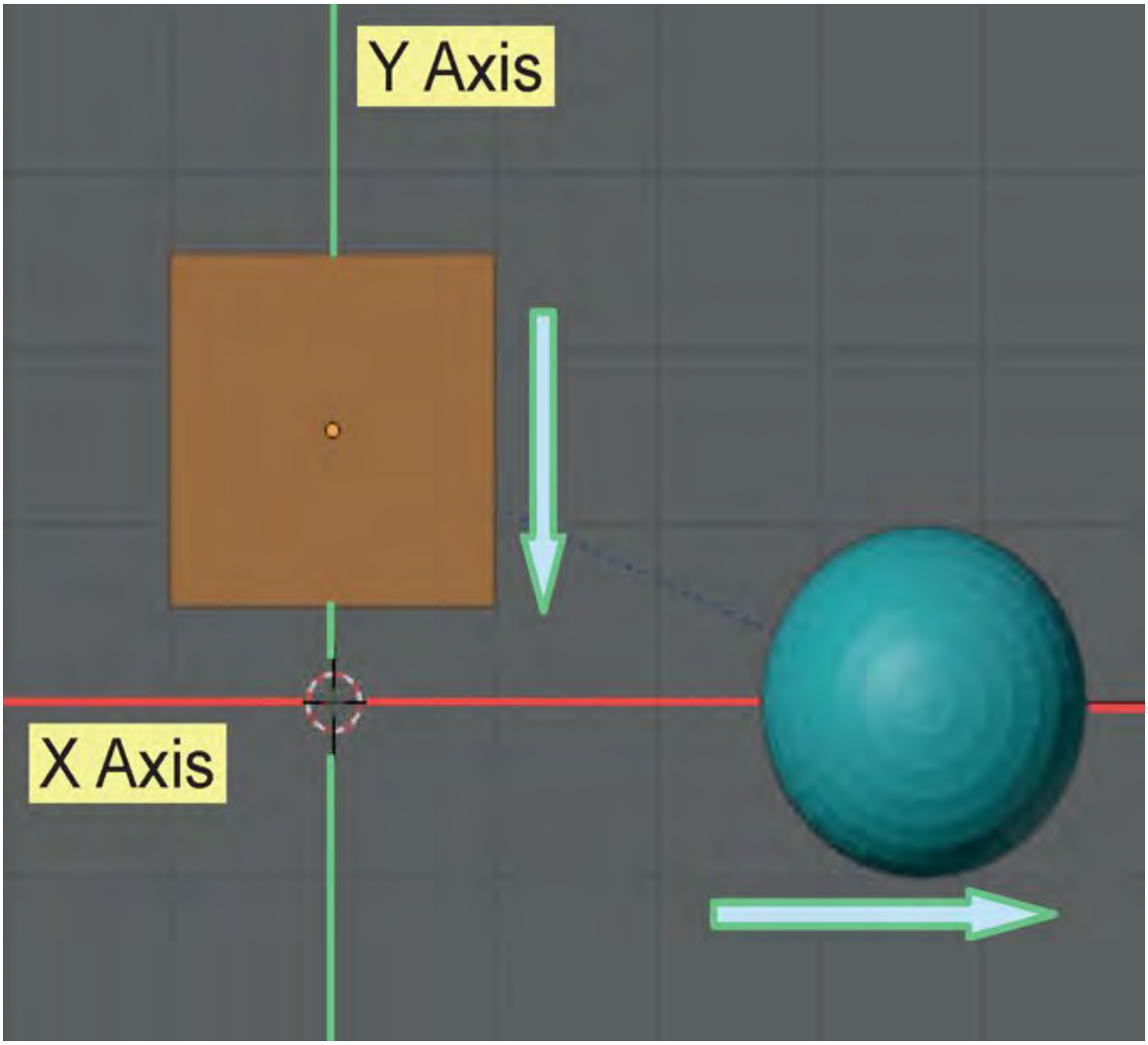


Figure 21.11

The **Map From,** Location values are retained per the previous example but **Note: the Map To, Location values** are changed. In particular **Note:** the **Y Source Axis Y is changed to X (Min Om and Max -2m).**

To apply a logical thought process to the procedure think of the Map From values controlling the **Map To** values.

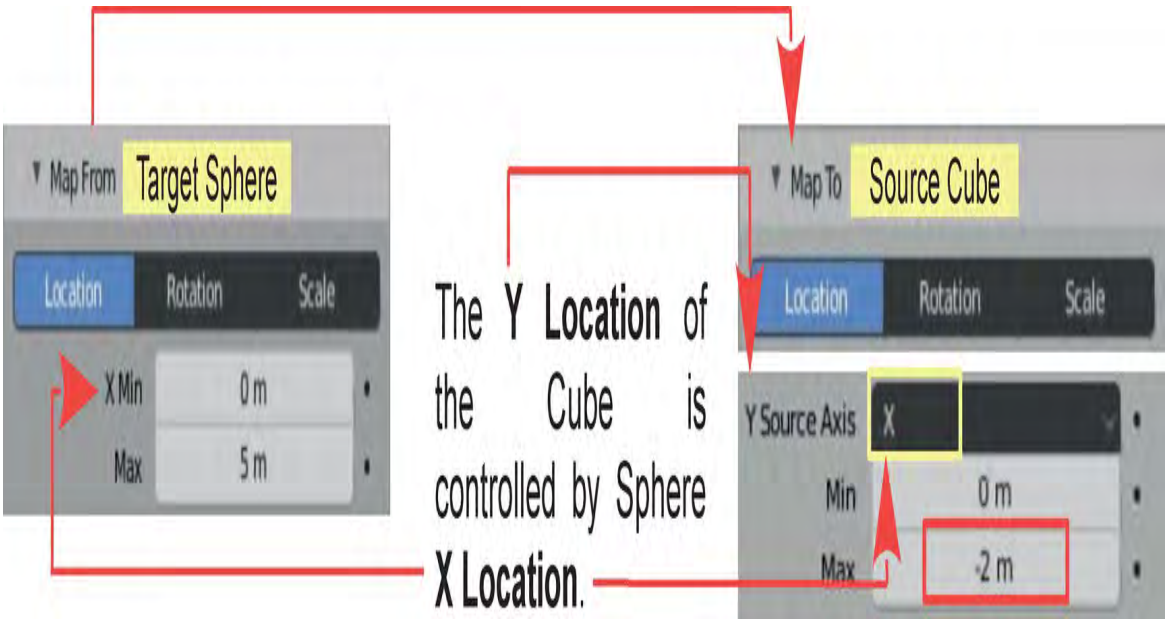


Figure 21.12

Location, Rotation or Scale Min – Max range values have to be entered in a Map From Axis channel and a Map To Axis channel, for a constraint to take effect.

As seen in Figure 21.12 the Range: Min 0 – Max 5m is entered in the Map From, Location channel. In the Map To the Range: Min 0m – Max **-2m** is entered in the Y Source Axis channel and, importantly, the **Y Source Axis** is set to be controlled by the **X Location channel.**

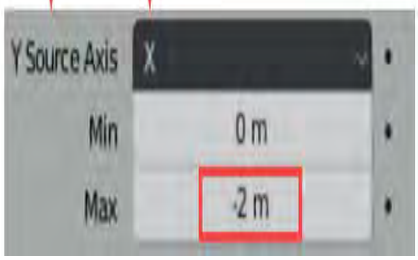


Figure 21.13

To reinforce the concept of the procedure, the **Translation** of the Sphere will control the **Rotation** of the Cube.

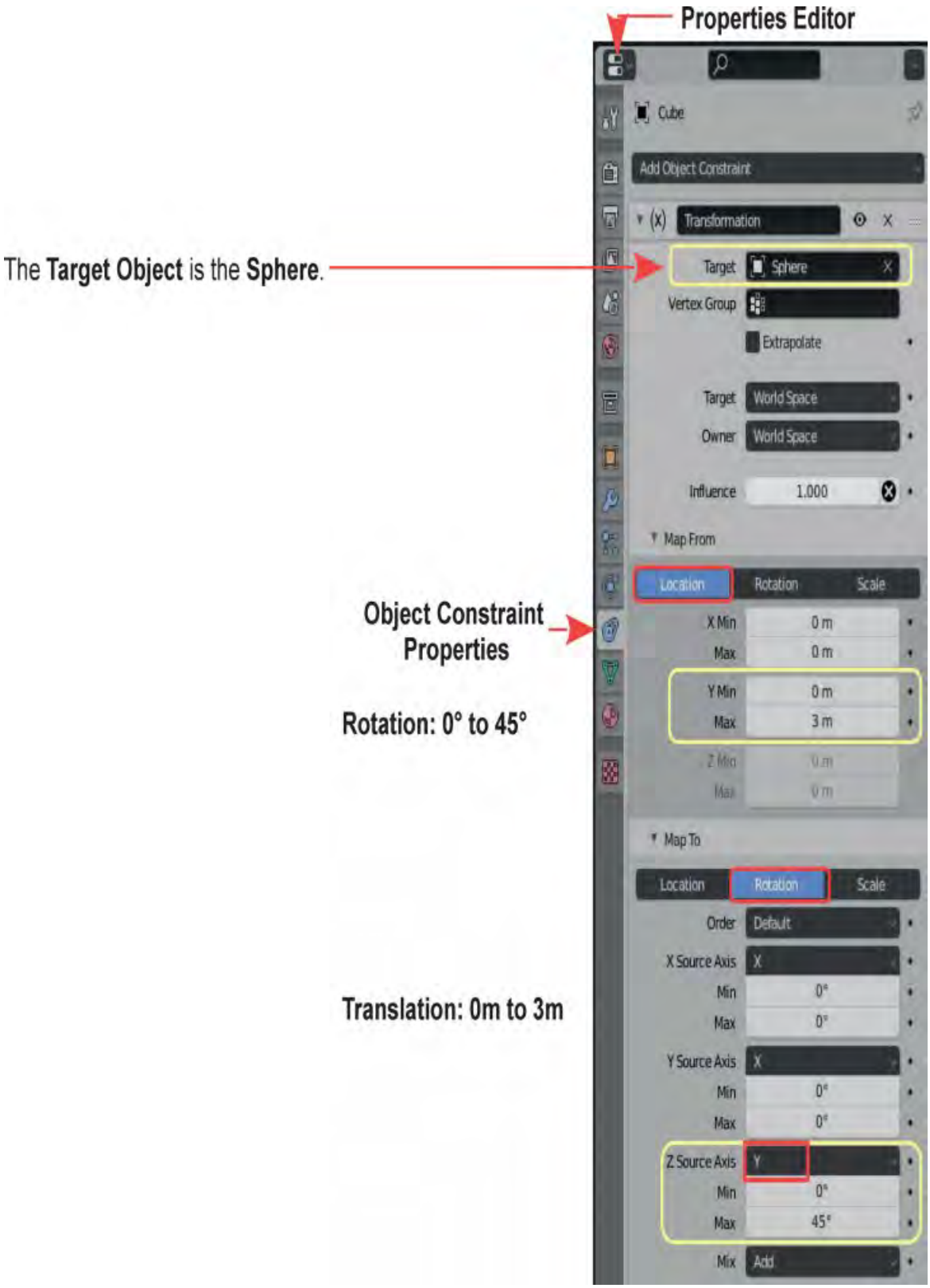

Properties Editor
The Target Object is the Sphere.
Object Constraint Properties
Rotation: 0° to 45°
Translation: 0m to 3m
Cube
Add Object Constraint
Transformation
Target
Sphere
Vertex Group
Extrapolate
Target
World Space
Owner
World Space
Influence
1.000
Map From
Location
Rotation
Scale
X Min
0 m
Max
0 m
Y Min
0 m
Max
3 m
Map To
Location
Rotation
Scale
Order
Default
X Source Axis
X
Min
0°
Max
0°
Y Source Axis
X
Min
0°
Max
0°
Z Source Axis
Y
Min
0°
Max
45°
Mix
Add

Figure 21.14

Important: Figure 21.15 shows the Objects in Top Orthographic View, therefore, the Z Axis of the Scene is normal to the computer Screen. The Axis of Rotation of the Cube is about the Z Axis.

The Map From (Target Sphere) Location Range is set for the Y Axis (Min Om – Max 3m) which means that the Sphere must be Translated on the Y Axis of the View.

The Map To (Source Cube) Range is set for Rotation about the Z Axis (Z Source Axis) being controlled by the values set in the Map From Y Axis Range.

Note: The use of Plus and Minus Values when setting Range limits.

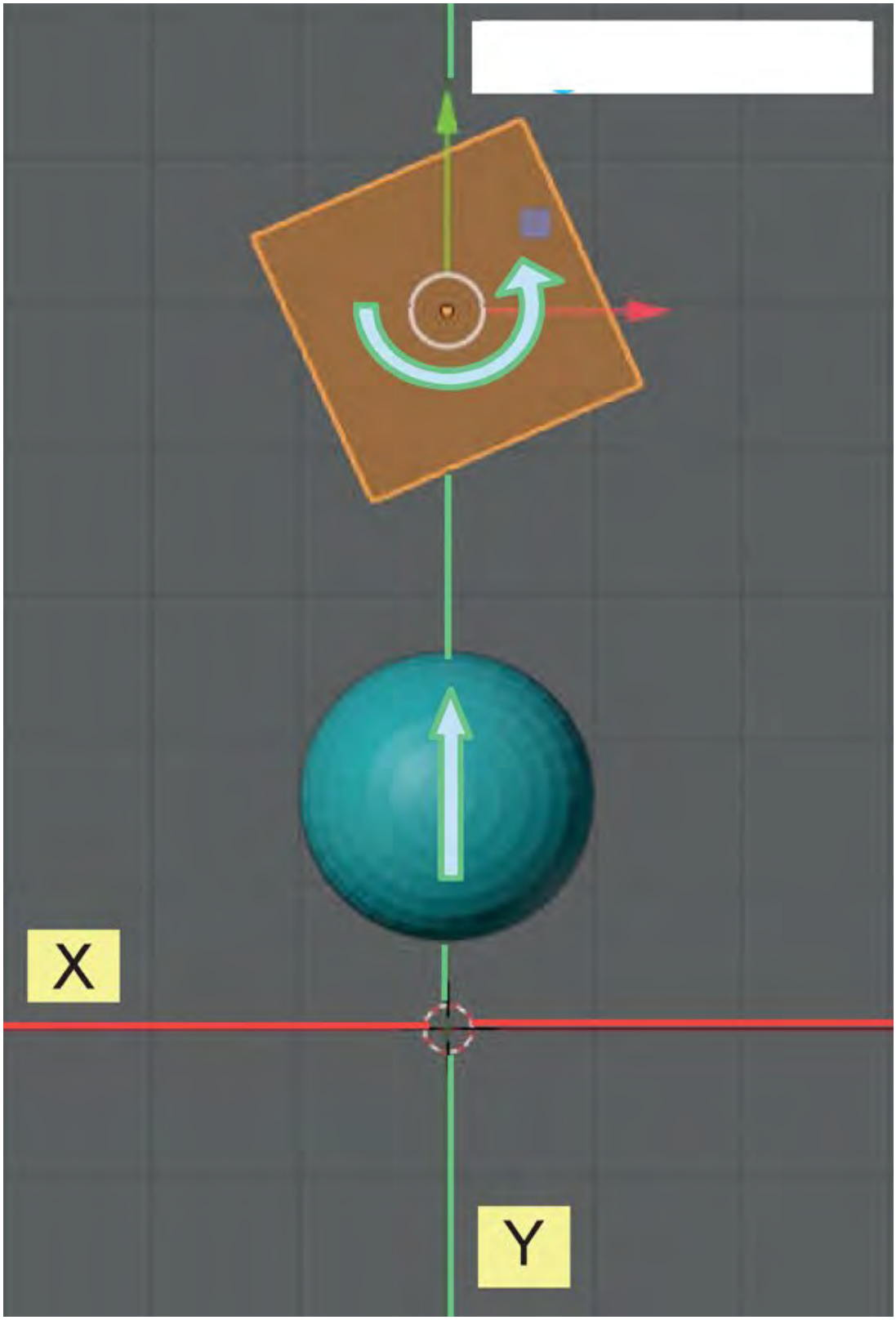


Figure 21.15

21.6 Tracking Constraints

- **Clamp To:** Clamps or locks the position of the Object to a target curve. Figure 21.16

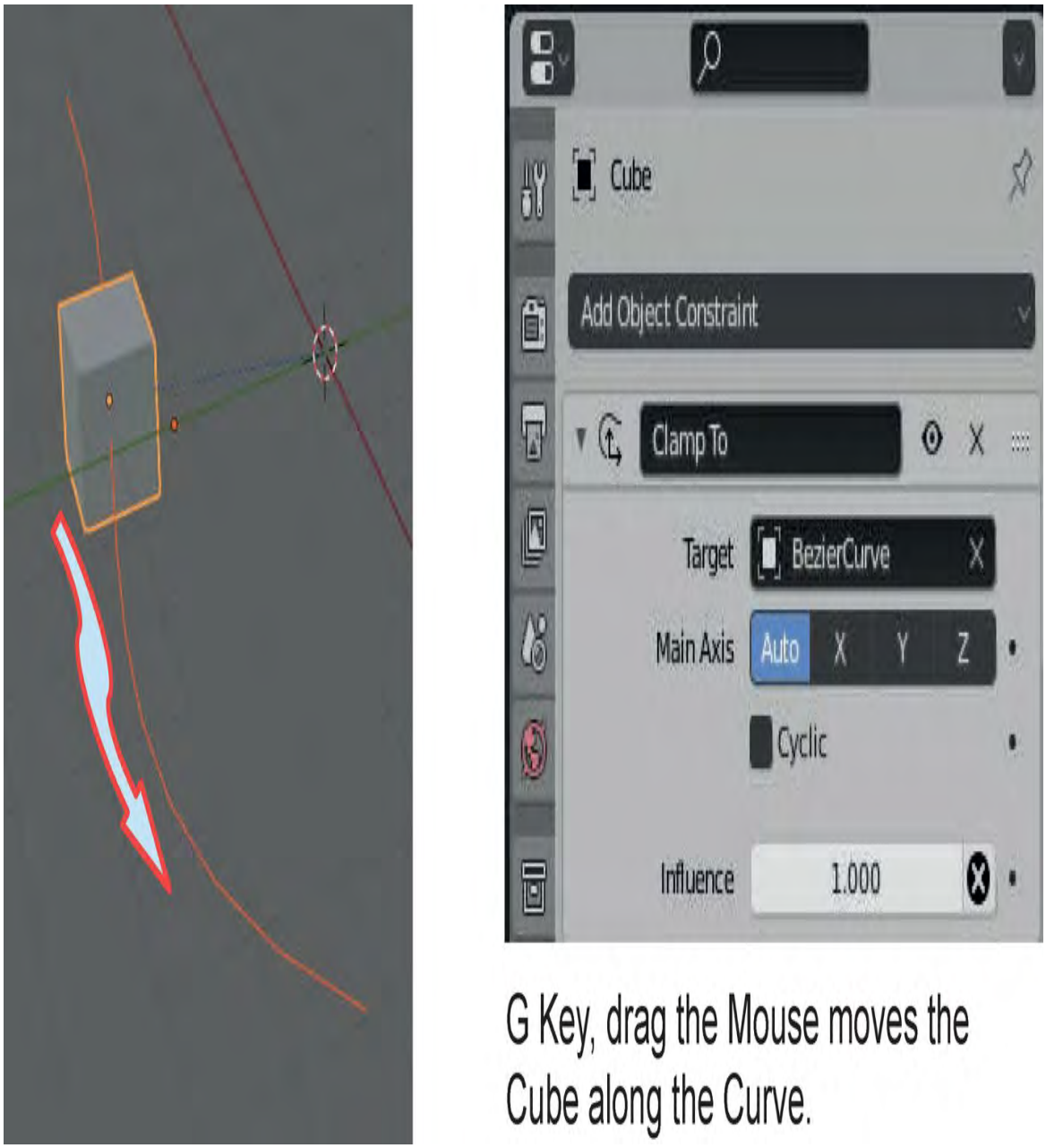
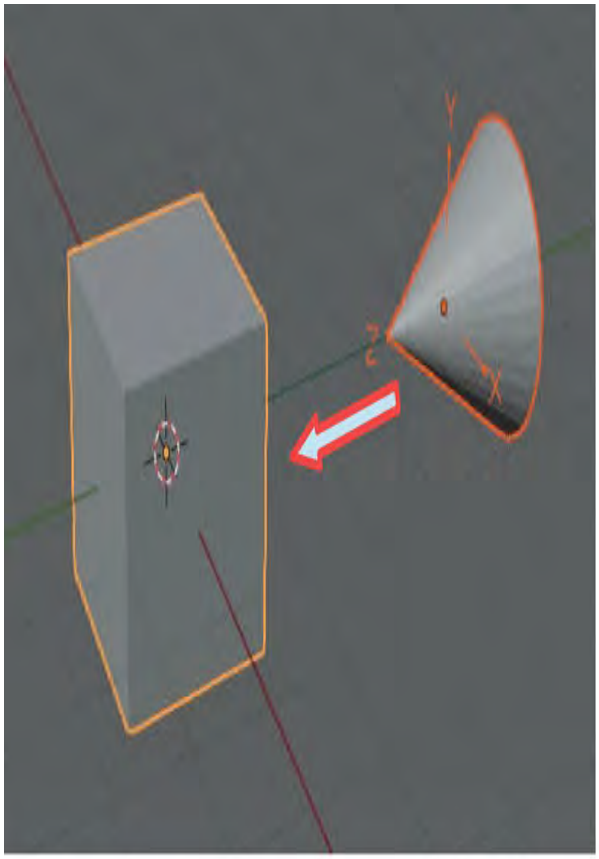


Figure 21.16

- **Damped Track:** Constrains one local axis of the Object to always point towards the target Object.



The Local Z Axis of the Cone points to the Cube.

Figure 21.17

- **Locked Track:** Similar to a Damped Track Constraint with more axis control.
- **Stretch To:** Stretches the Object towards the Target Object or compresses the Object away from the Target Object.
- **Track To:** As seen in the introduction to Constraints the Track To Constraint causes the Object to always point towards the Target Object no matter where

either the Object or the Target is positioned (Figure 21.4 Camera points to the Cube).

21.7 Relationship Constraints

- **Action:** See Section 21.8.
- **Child Of:** Chapter 22 – 22.3.
- **Floor:** Allows the Target Object to obstruct the movement of the Object. For example, a Sphere animated to descend in a Scene will not pass through a Plane that has been set as a Target Object.
- **Follow Path:** Causes the Object to be animated to follow a Curve Path nominated as the Target. This Constraint also has the feature to follow the Curve, which means that the Object will rotate and bank as it follows the Curve. This constraint can also be employed to duplicate Objects along a Curve Path. (See Chapter 18 – 18.18).
- **Pivot:** Causes the Object to leapfrog to the opposite side of the Target Object along an axis between the Object and the Target Center. The location can be offset on either side of the axis by inserting offset values.
- **Shrinkwrap:** Locks an Object to the surface of another mesh Object that is set as the Target.

21.8 The Action Constraint

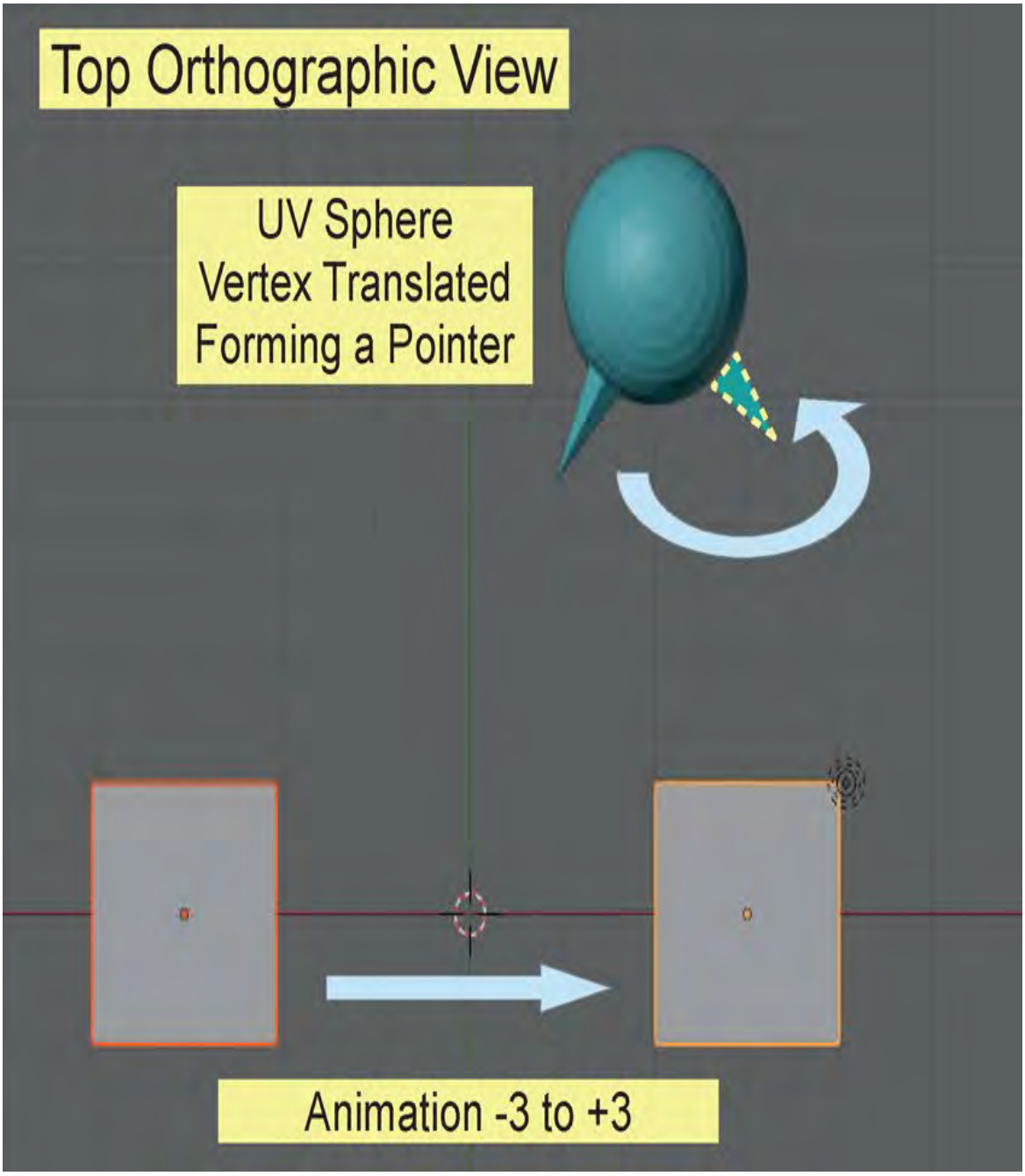


Figure 21.18

The **Action Constraint** allows you to control the action of one Object by manipulating the action of another. An action may be the Translation, Rotation, or Scale of an Object. To demonstrate, the **Rotation of a Sphere Object will control the Translation of a Cube.**

The location of the UV Sphere in the Scene is not important.

Animate the Cube to move from minus three Blender units to plus three units on the X Axis in 100 Frames (see Chapter 20 for Animation). Place the animation at Frame 1.

Select the **Cube** then in the **Properties Editor, Constraints Properties,** Add **Relationship Constraint** type **Action.** In the Constraint panel set the values as shown in Figure 21.9 (over).

Note: On entering the **Action** as; **CubeAction** in the **Action Tab** the Cube jumps from position minus three to minus six. When the Animation is played the Cube moves from minis six to the center of the Scene. Rotating the Sphere about the Z Axis will Translate the Cube

The Z Rotation of the Target (UV Sphere) controls the Action of the Cube within the limits of the Animation (the movement set in 100 Frames).

The Z Rotation (Target Range) is limited to: 0 to 90° (see Figure 21.9 over).

Cube
Add Object Constraint
Action
Target
Sphere
Vertex Group
Evaluation Time
0.000
Mix
After Original
Influence
1.000
Target
Channel
Z Rotation
Target
World Space
Range Min
0.000
Max
90.000
Action
Action
CubeAction
Object Action
Frame Start
1
End
100

Figure 21.19

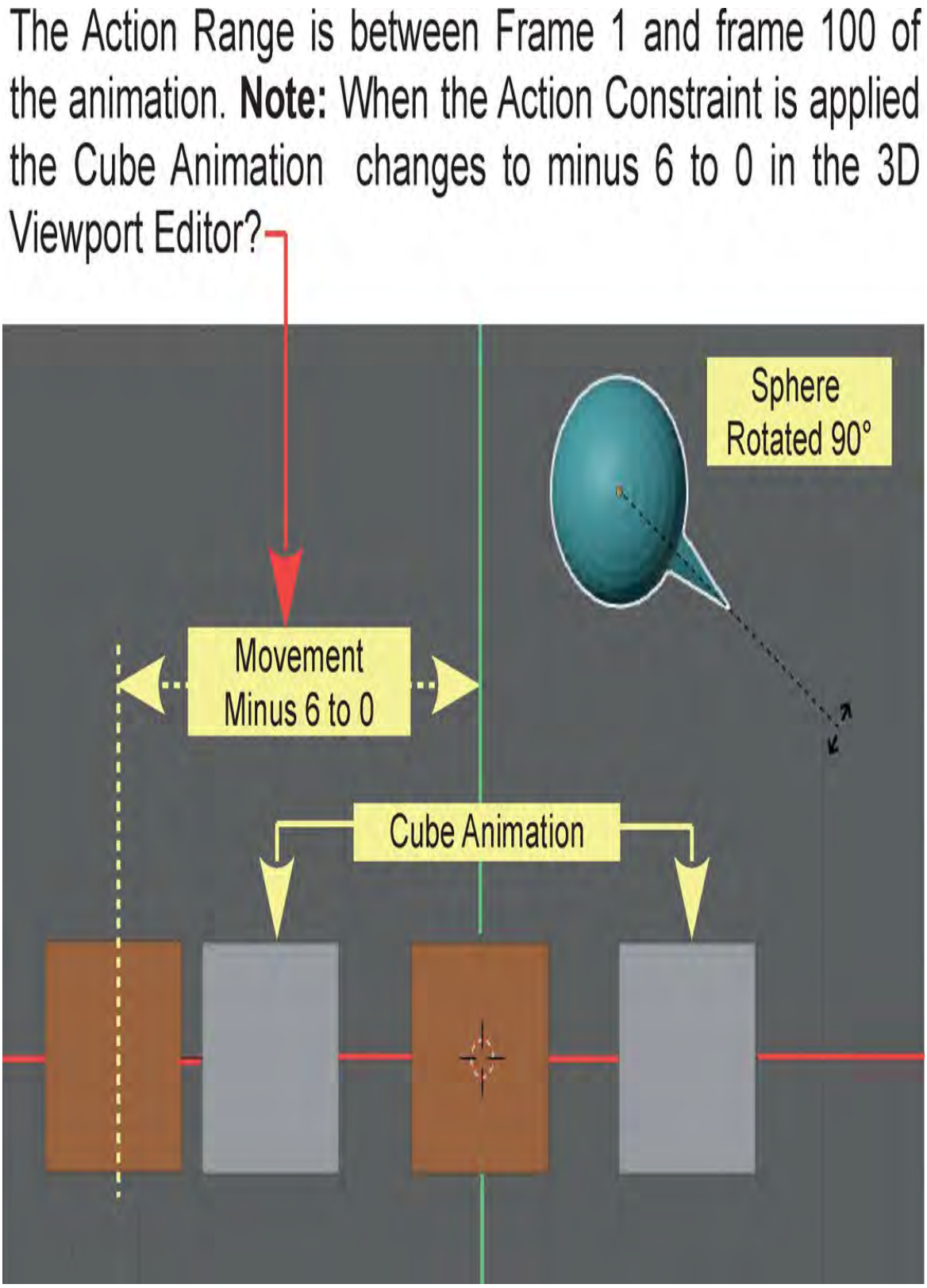

The Action Range is between Frame 1 and frame 100 of the animation. **Note:** When the Action Constraint is applied the Cube Animation changes to minus 6 to 0 in the 3D Viewport Editor?
Sphere Rotated 90°
Movement Minus 6 to 0
Cube Animation

Figure 21.20

You may set an Animation to rotate the UV Sphere in which case when the Cube is Translated (Moved) the Sphere Rotates.

21.9 The Shrinkwrap Constraint

The **Shrinkwrap Constraint** could be more aptly named the **Mesh Surface Lock** since the constraint locks an Object to the surface of another mesh Object that is set as the **Target.**

Do not confuse this constraint with the **Shrinkwrap Modifier.** To demonstrate how the constraint operates; In the default Blender Scene, in Front Orthographic view, add a **UV Sphere** and an **Icosphere.** Scale the **UV Sphere** way down, and arrange the objects as shown in Figure 20.21.

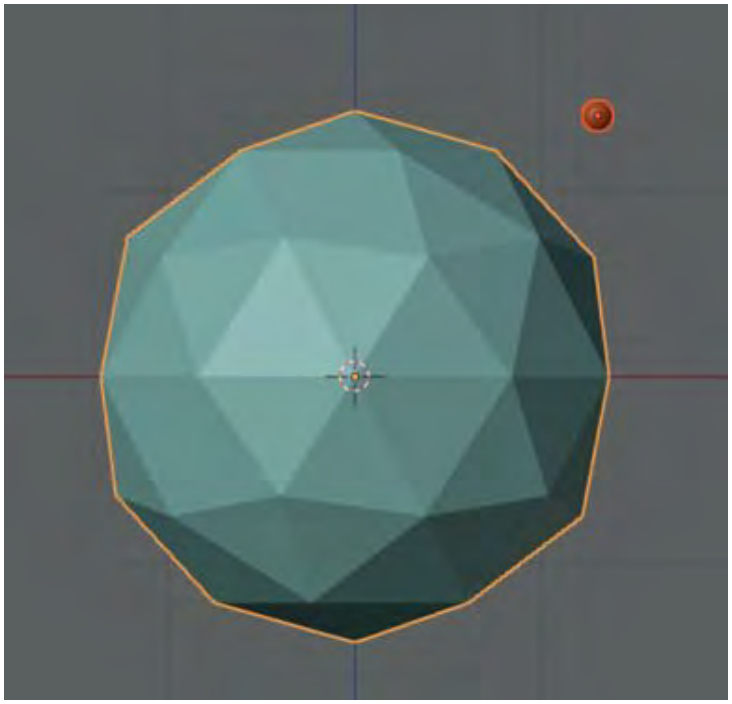


Figure 21.21

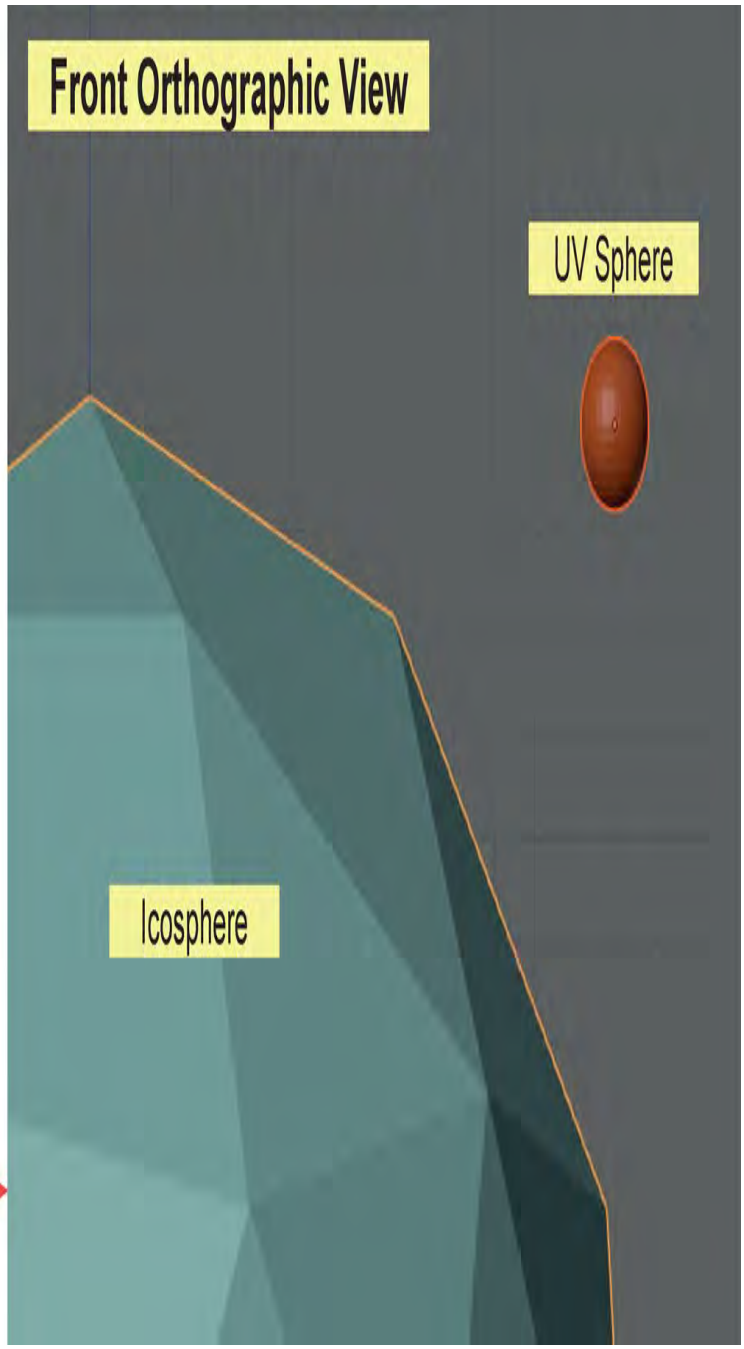


Figure 21.22

Select the **UV Sphere** and in the **Properties Editor, Object Constraints** add a **Constraint**

(Figure 20.23).

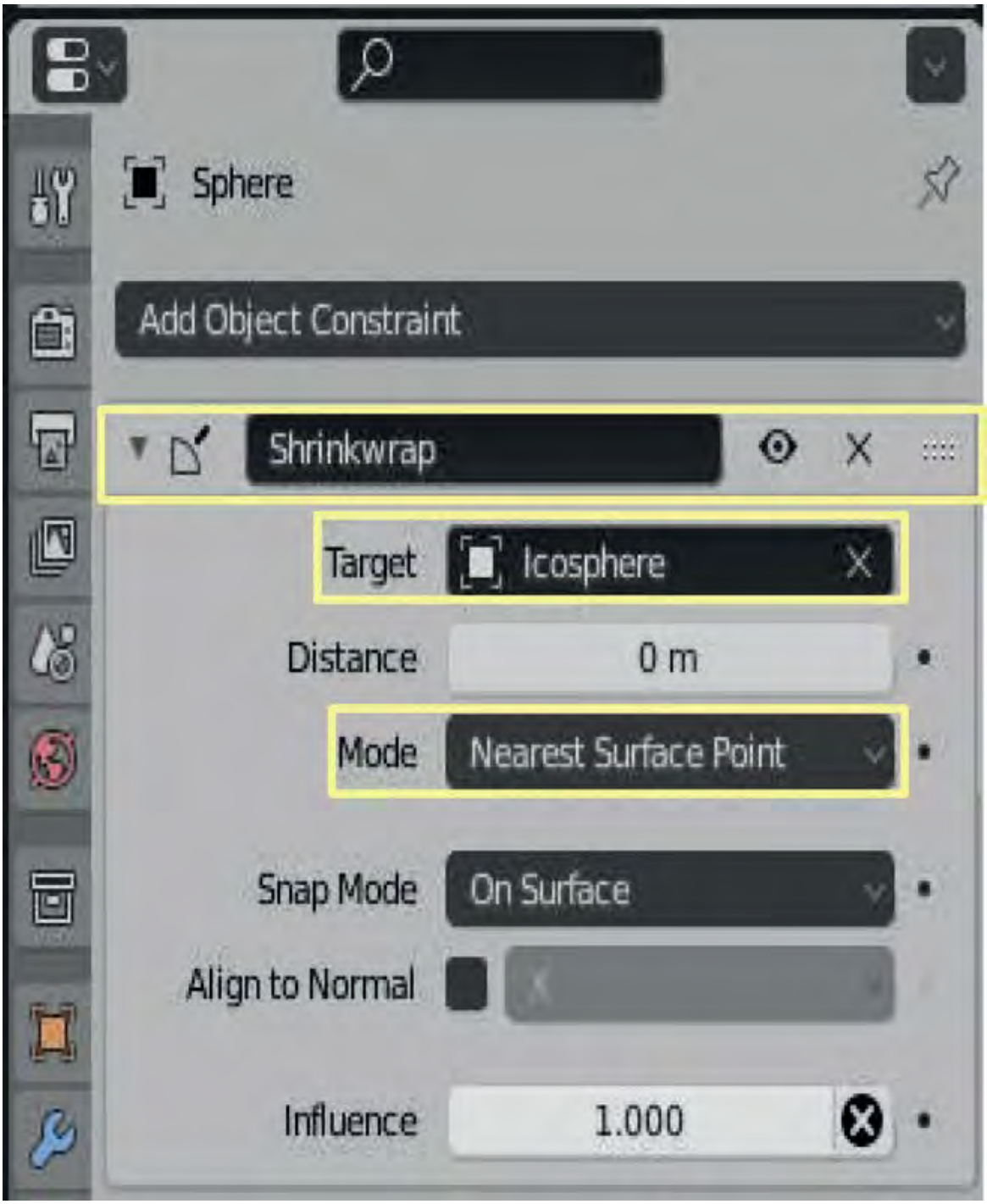


Figure 21.23

In the **Shrinkwrap Constraint panel,** click in the **Target selection bar** and select **Icosphere** as the Target (Figure 20.23).

In the default make note **that the Mode** is **Nearest Surface Point.**

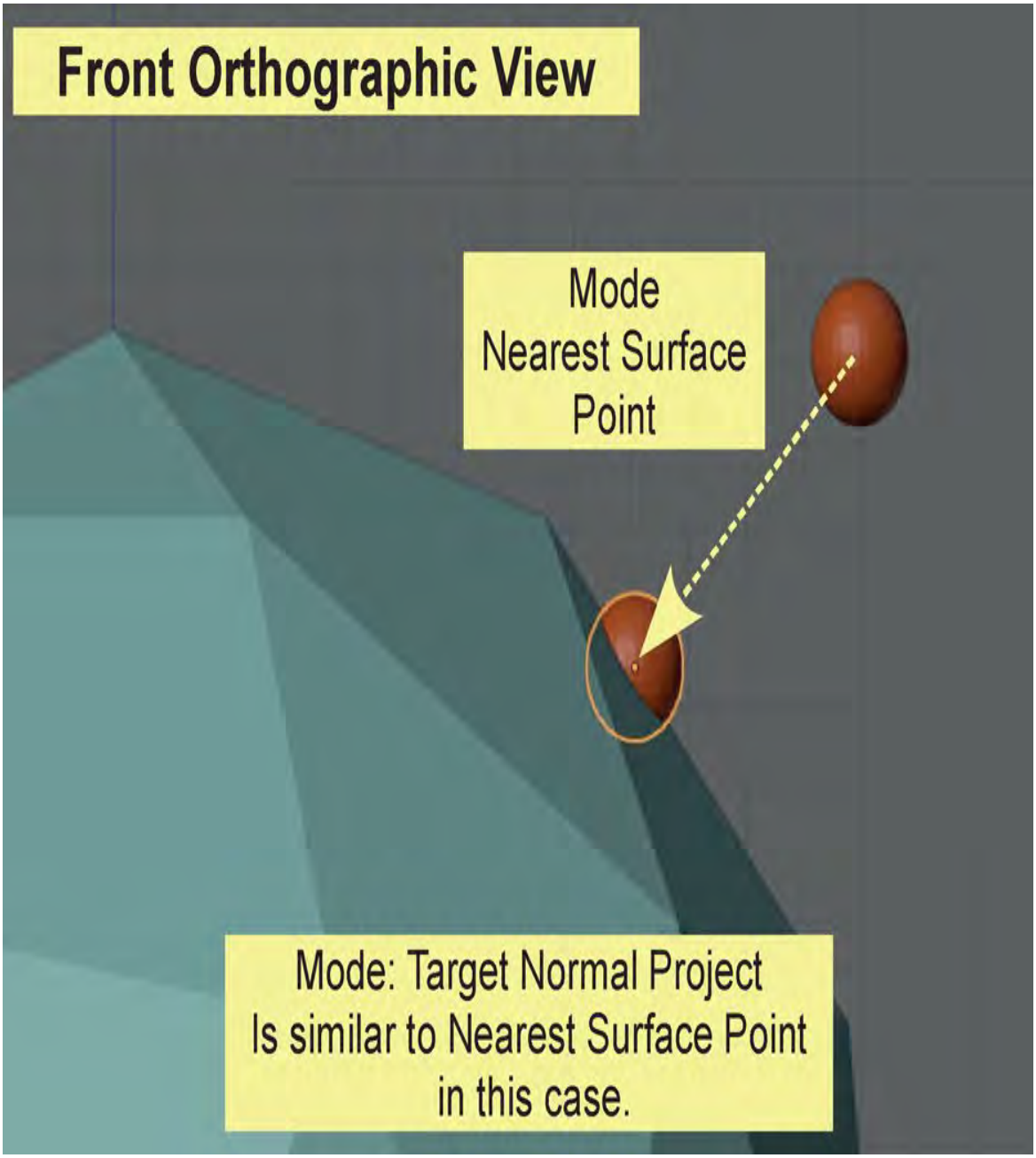

Front Orthographic View
Mode
Nearest Surface
Point
Mode: Target Normal Project
Is similar to Nearest Surface Point
in this case.

Figure 21.24

Figure 21.25

With the **Mode: Nearest Surface Point** and the **Target** set as **Icosphere** the UV Sphere relocates to the surface of the Icosphere at the nearest point to the original position of the UV Sphere (Figure 20.24). **Mode: Nearest Surface Point** is one of four options (Figure 20.25).

Mode: Project.

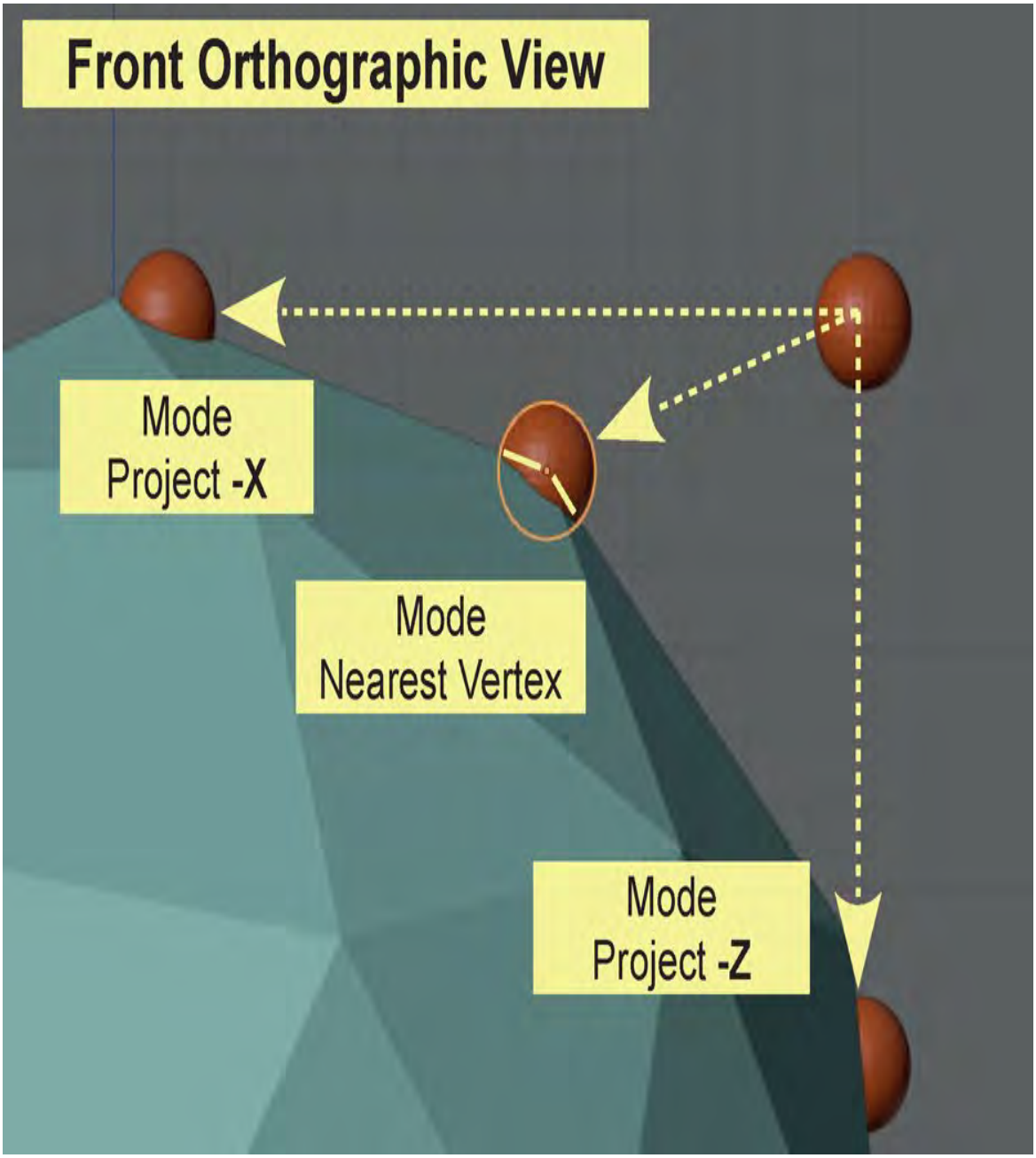


Figure 21.26

With Mode: Project selected, **Axis X, Axis Y, and Axis Z** buttons are present in the **Constraints panel** (Figure 20.27 **Note** there are positive and negative values).

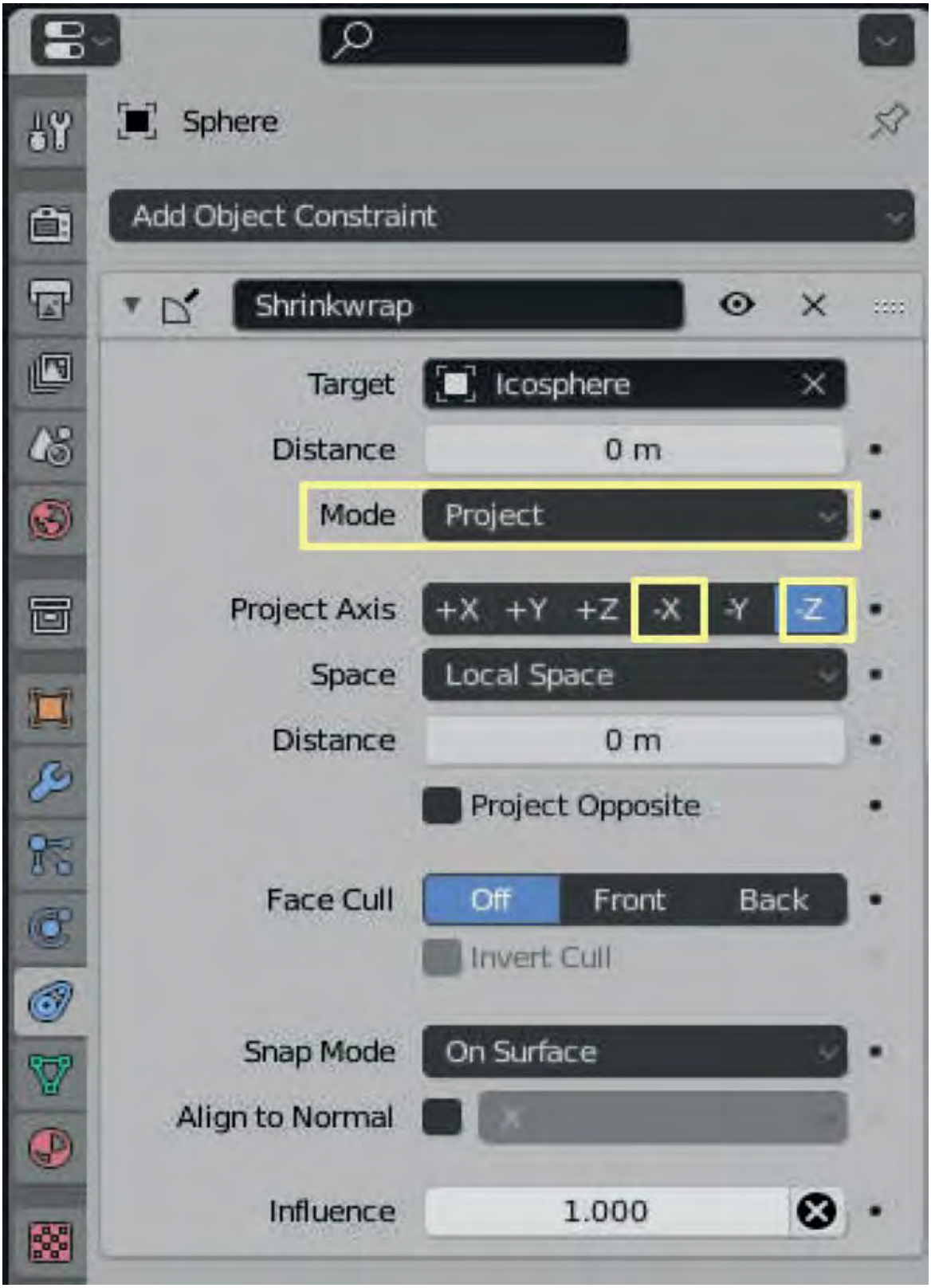

Sphere
Add Object Constraint
Shrinkwrap
Target
Icosphere
Distance
0 m
Mode
Project
Project Axis
+X
+Y
+Z
-X
-Y
-Z
Space
Local Space
Distance
0 m
Project Opposite
Face Cull
Off
Front
Back
Invert Cull
Snap Mode
On Surface
Align to Normal
Influence
1.000

Figure 21.27

Note: In the Shrinkwrap Constraint Properties, the **Mode is: Project** and **Space is: Local Space**

Local Space means that the **Projection** is determined by the **Local Axis** of the selected Object.

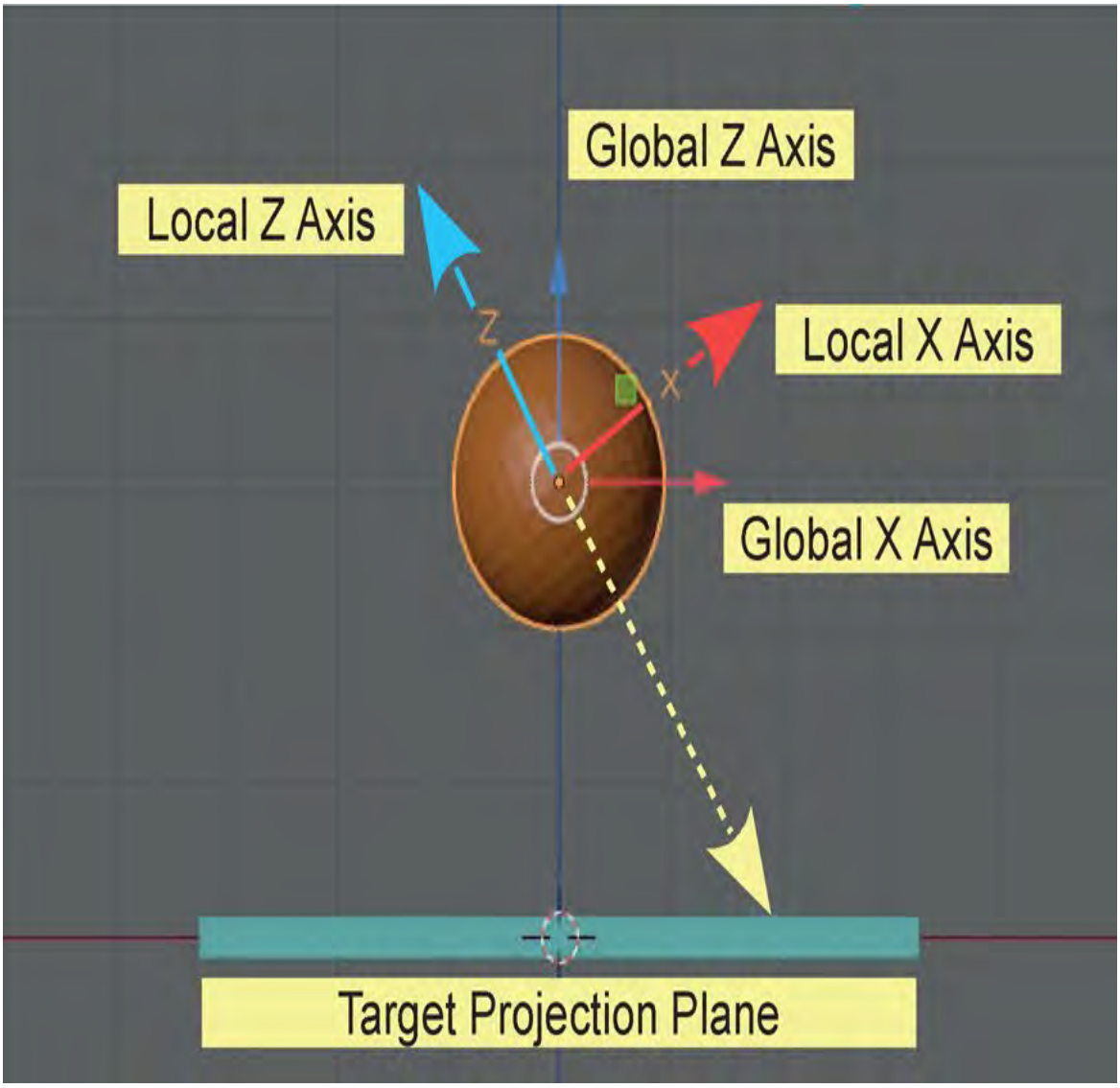


Figure 21.28

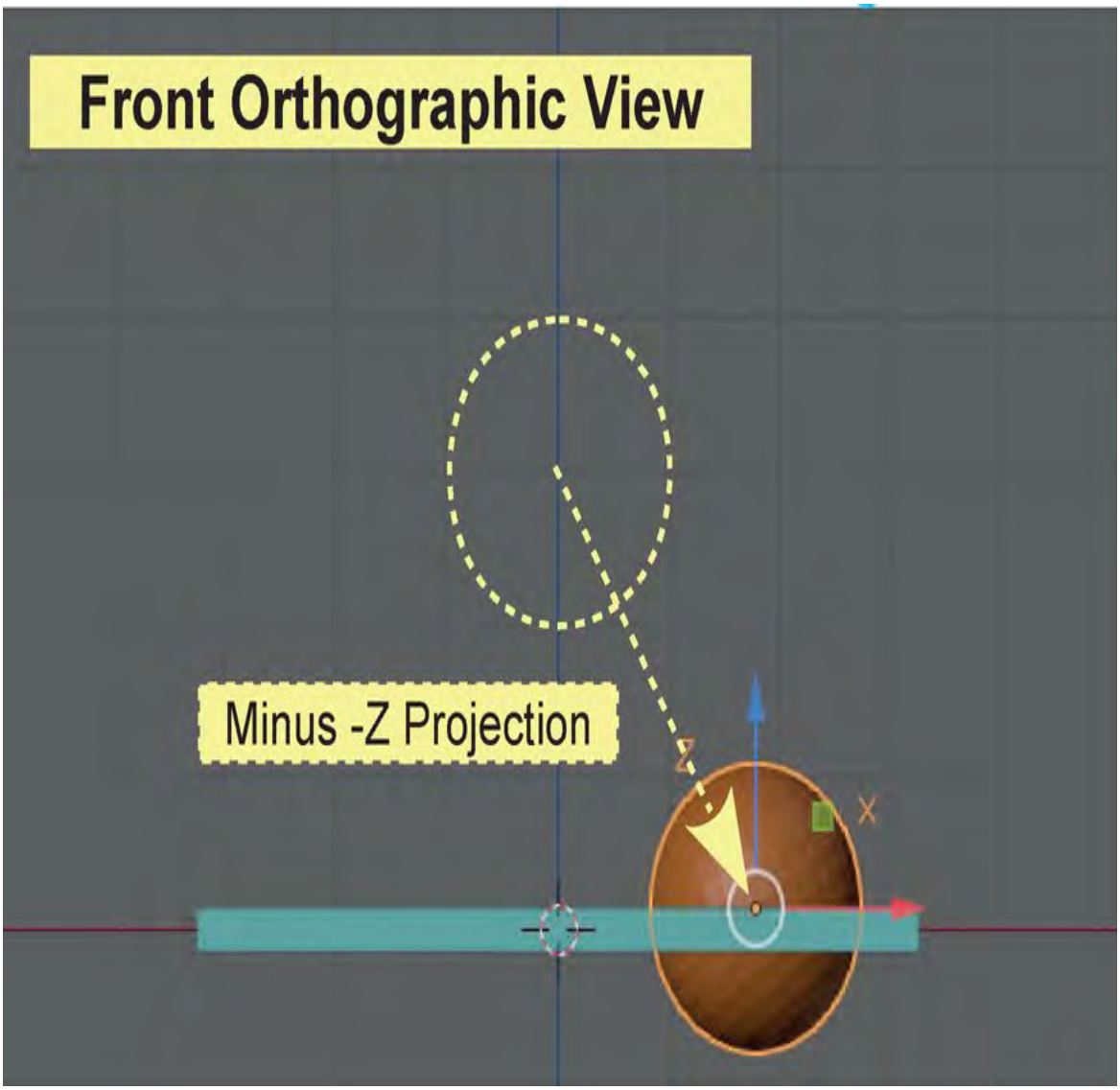


Figure 21.29

Figures 21.28 shows the difference between the Local Axis of the UV Sphere when the Sphere is Rotated. Since the Local Minus -Z Axis is set as the Project Axis in the Constraint Properties the Sphere is projected to the nearest point on the Target Projection Plane along its **Minus Z Axis.**

Pressing the **R Key** and Rotating the Sphere alters the direction of the Minus Z Axis causing the Sphere to be moved along the surface of the Target Projection Plane. (Figure 21.30).

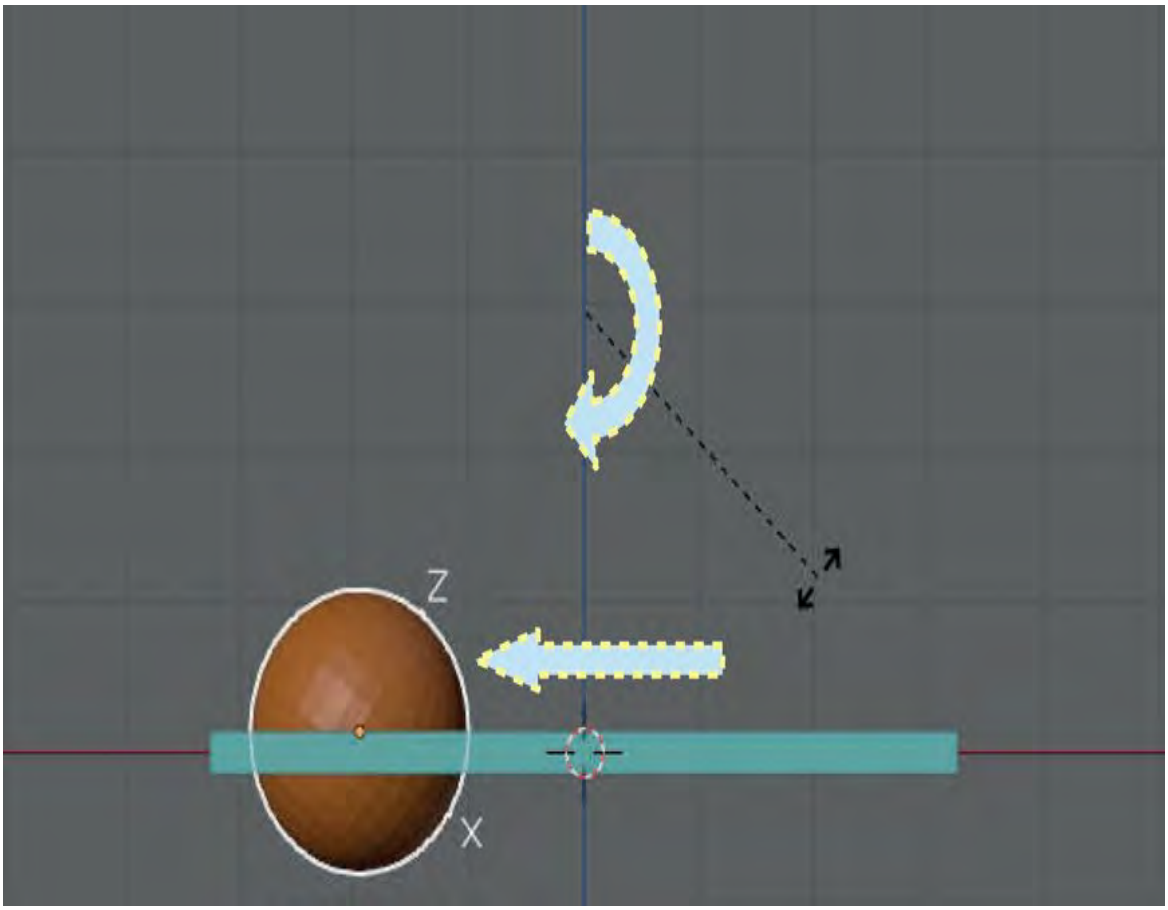


Figure 21.30

21.10 Follow Path – Extrude/Bevel

The **Follow Path Constraint** causes an Object to follow a **Curve Path** set as the **Target when the Object is Animated to move.** The constraint incorporates a **Follow Curve** setting which make the Object rotate and bank as it moves along the Path in the Animation. This has been demonstrated in Chapter 20-20.18 **Animation Follow Path.**

The **Follow Path Constraint** may also be used to **Generate** shapes by extruding a profile along the length of a Curve Path. Consider that a three dimensional regular shape is defined by its cross section profile and the curvature of its length.

Review Chapter 10 and in particular 10.6 Modeling from a Curve where a Bezier Curve was manipulated to generate a Mesh Object (Figure 10.15). The Mesh Object created was a regular shape. The following procedure will

demonstrate how **irregular shapes** may be created by generating a profile along a Curve Path.

The Cross Section Profile and Length

Figure 21.31 shows a Cross Section Profile and a Length Curve

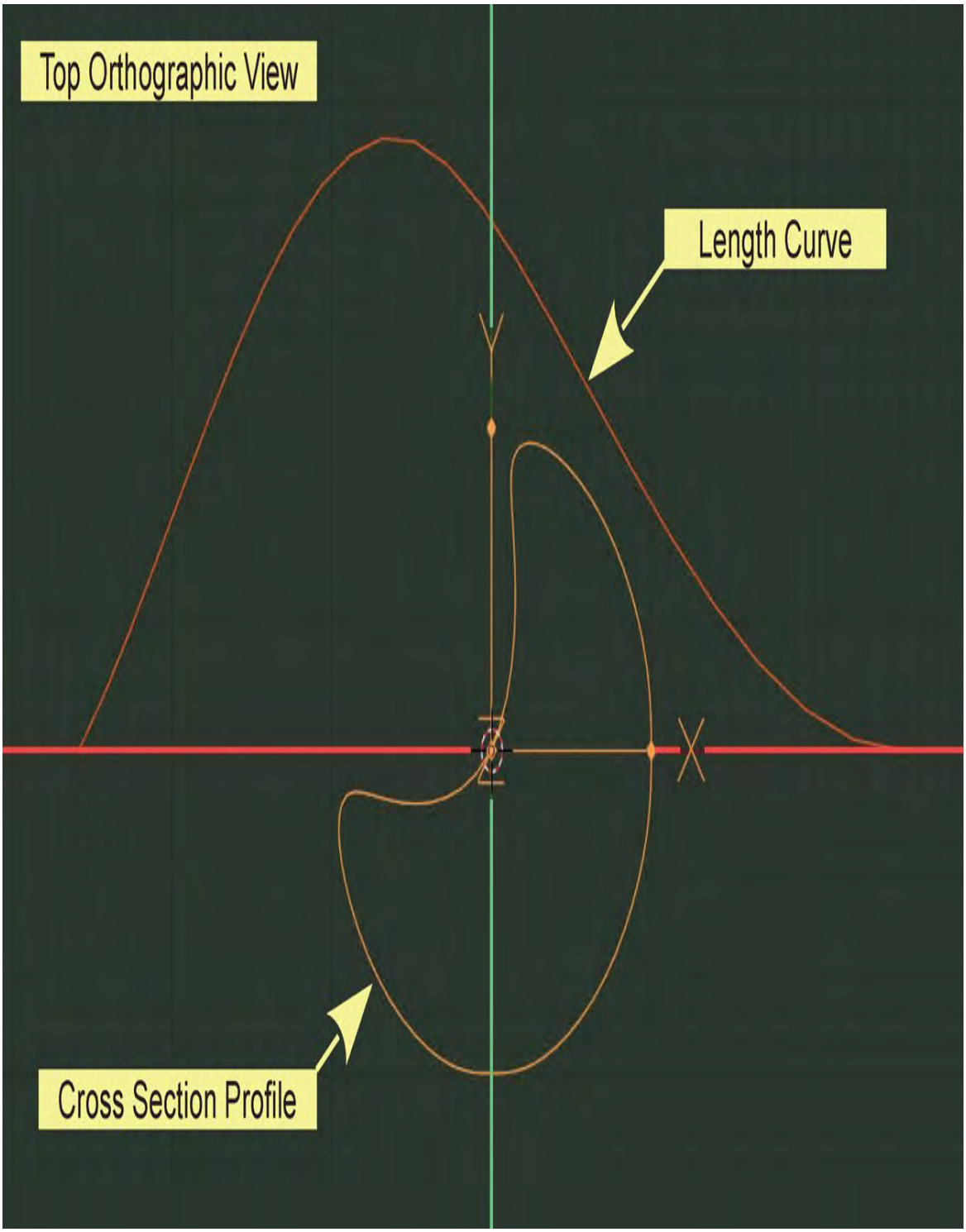


Figure 21.31

The Cross Section Profile is a **Nurbs Circle** reshaped in Edit Mode and the Length Curve is a **Bezier Curve,** Scaled up and reshaped in Edit Mode. (see

Chapter 10 – 10.3, 10.5). The Length Curve (Bezier Curve) will be the centerline of the Extrusion.

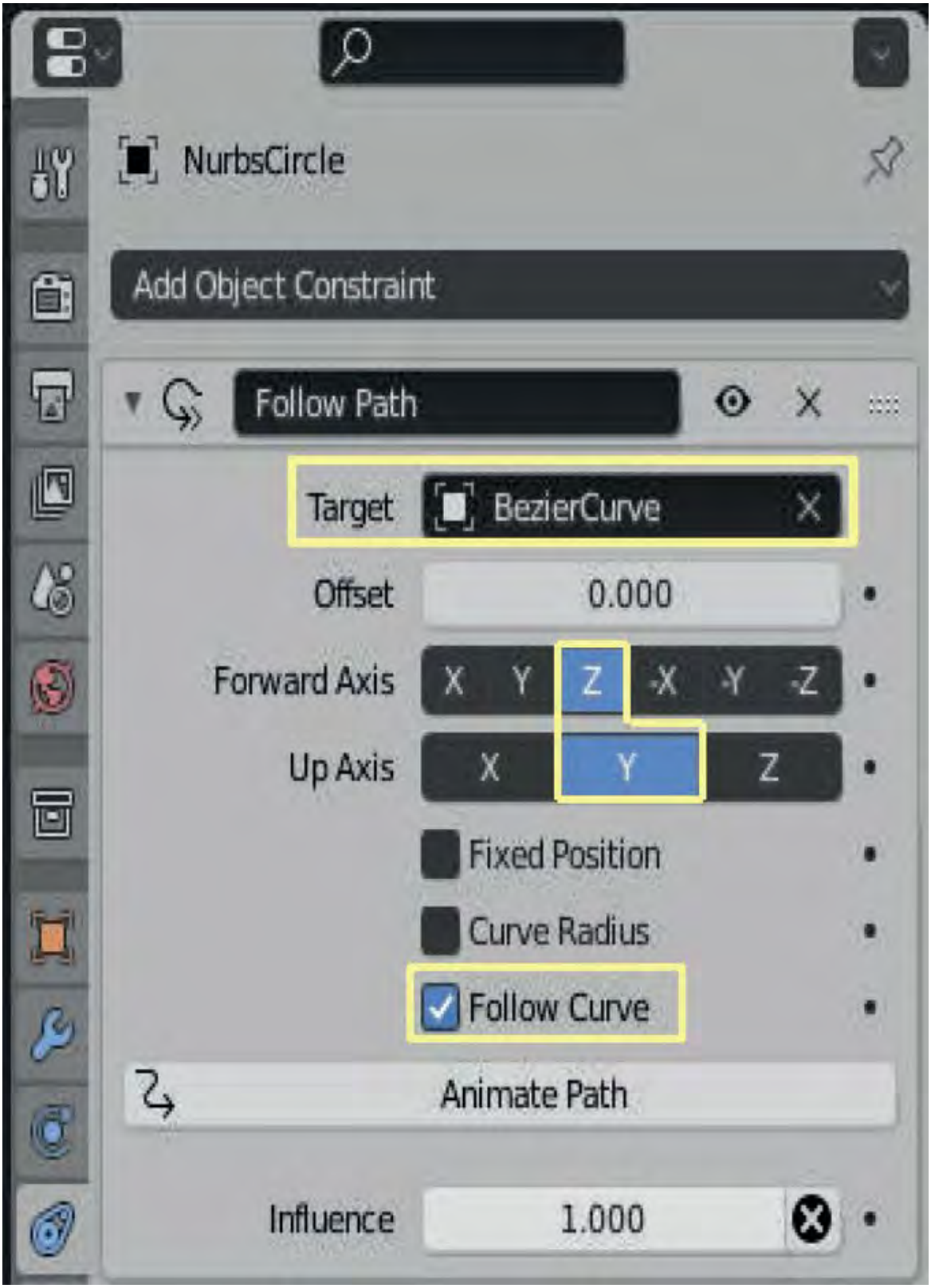

NurbsCircle
Add Object Constraint
Follow Path
Target
BezierCurve
Offset
0.000
Forward Axis
X
Y
Z
-X
-Y
-Z
Up Axis
X
Y
Z
Fixed Position
Curve Radius
Follow Curve
Animate Path
Influence
1.000

Figure 21.32

To Extrude the Cross Section along the Length Curve, the Cross Section has to be orientated and positioned at the start of the Length Curve. Note: Figure 21.31 shows the 3D Viewport Editor in Top Orthographic View, therefore, the Local Axis of the Profile coincides with the Global XY Axis of the Scene.

To orientate and position the Cross Section for Extrusion, select the Cross Section (Nurbs Circle – Modified) and in the **Properties Editor, Object Constraint Properties** add a **Follow Path Constraint** (Figure 21.32). Enter **Target** as **BezierCurve**. The Cross Section relocates to the start of the Bezier Curve (Figure 21.33).

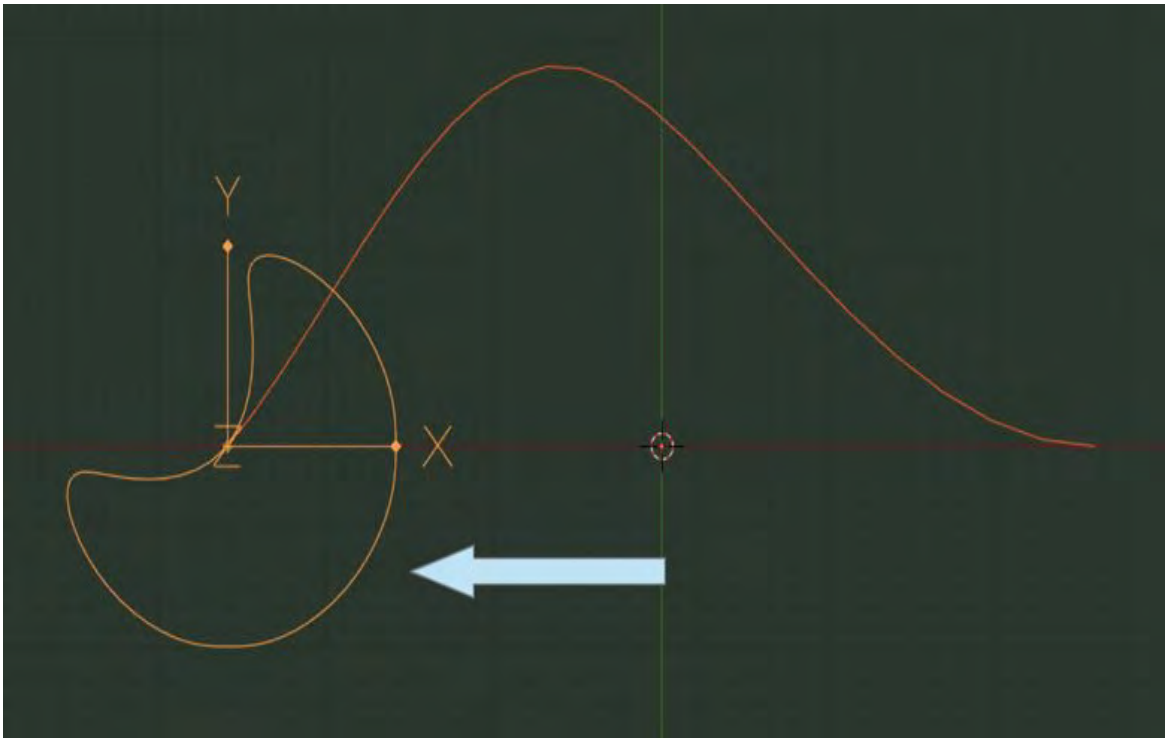


Figure 21.33

In the Follow Path Constraint the default Forward Axis is Y (highlighted blue) and the Up Axis is Z. This corresponds to the Local Axis of the Cross Section. To Extrude the Cross Section along the Length Curve the Cross Section is required to be at right angles to the Length Curve.

To Rotate the Profile, in the Properties Editor, Follow Path Constraint, change the Forward Axis to **Z** and the Up Axis to **Y. No change occurs** in the orientation until you check **Follow Curve** in the Constraint panel.

Figure 21.34

With **Follow Curve** checked you will see the Profile at the end of the Curve Path with its Local Axis; Y Up and Z Forward pointing approximately in line with the Curve Path (Figure 21.34).

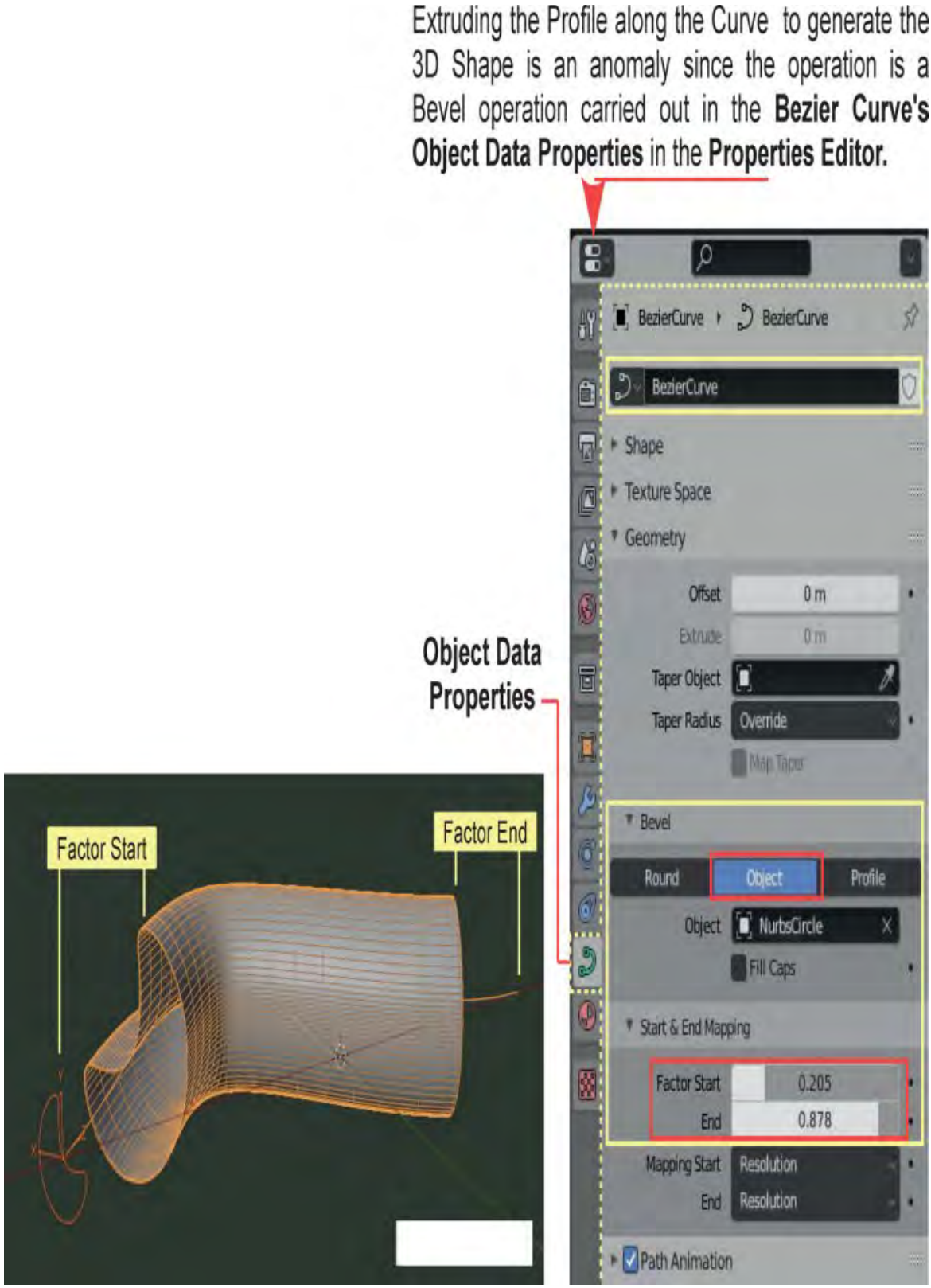

Extruding the Profile along the Curve to generate the 3D Shape is an anomaly since the operation is a Bevel operation carried out in the **Bezier Curve's Object Data Properties** in the **Properties Editor.**
Object Data Properties
Factor Start
Factor End
BezierCurve
BezierCurve
BezierCurve
Shape
Texture Space
Geometry
Offset
0 m
Extrude
0 m
Taper Object
Taper Radius
Override
Map Taper
Bevel
Round
Object
Profile
Object
NurbsCircle
Fill Caps
Start & End Mapping
Factor Start
0.205
End
0.878
Mapping Start
Resolution
End
Resolution
Path Animation

Figure 21.35

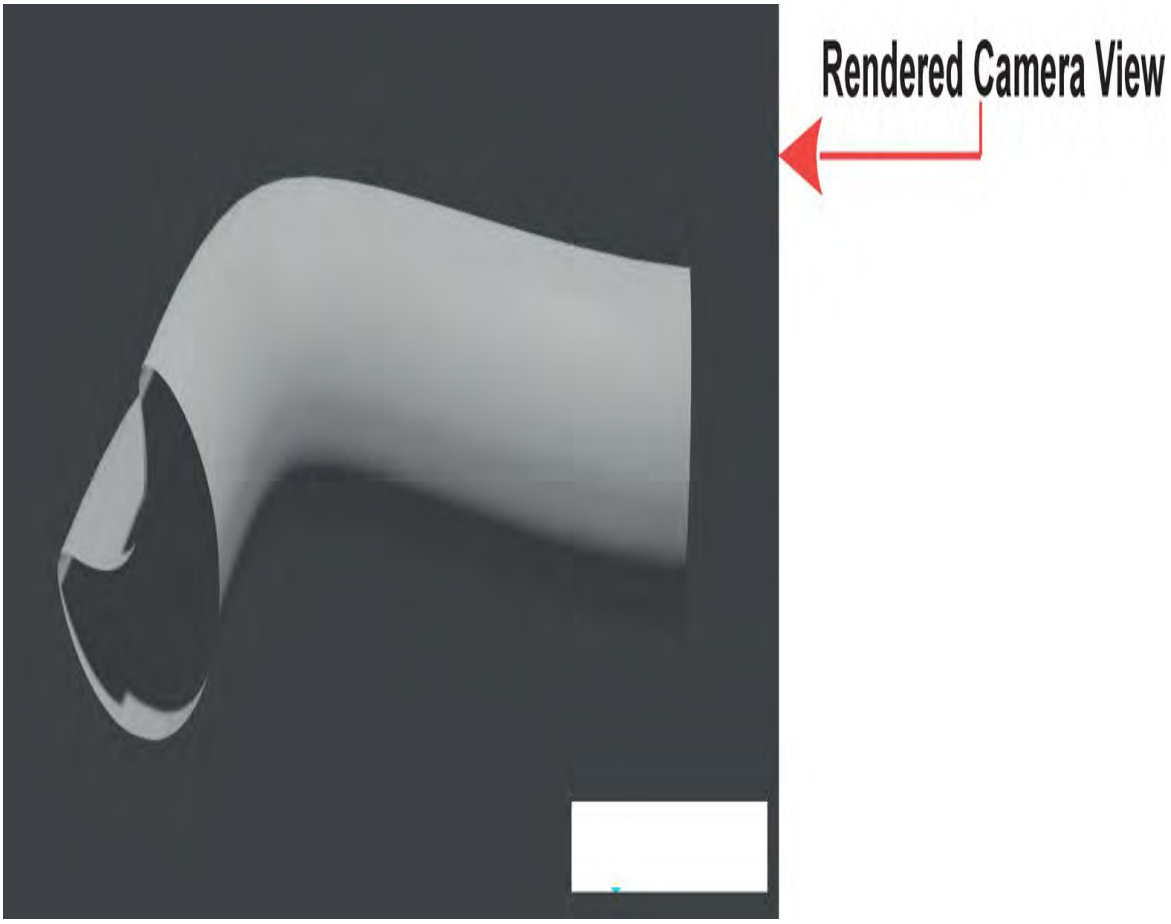


Figure 21.36

Scale the Nurbs Circle down before applying the Bevel to the Bezier Curve.

Note: The difference in orientation between the Nurbs Circle Profile and the profile of the final Object.

22 Armatures & Character Rigging

22.1 Single Bone Armature
22.2 Adding Armatures
22.3 Adding Bones
22.4 The Armature Rig
22.5 Multiple Armatures
22.6 Child Parent Relationship
22.7 Armature Display Types
22.8 Building an Armature Rig
22.9 The Extrude Tool
22.10 Deforming a Mesh
22.11 The Armature Modifier
22.12 Assigning Vertices – Groups
22.13 Assigning Vertices-Set Parent
22.14 Assigning Vertices-Weigh Paint
22.15 Inverse Kinematics Constraint
22.16 Spline IK Constraint
22.17 Forward
22.18 Character Rigging
22.19 Bone Naming
22.20 Assigning the Mesh
22.21 Vertex Groups
22.22 Posing the Character Model
22.23 Pre-Assembled Armatures

Armatures are used to control the movement of Objects or components of an Object, in animation. In intricate assemblies, components are linked or associated with Armatures, such that, when the Armature moves the components move. Armatures themselves are made up from parts called **Bones** and may consist of a single Bone or multiple Bones linked in **Child Parent Relationships.** This means that when one Bone moves other Bones move according to the hierarchy in the relationship. Armatures do not Render, therefore, Bones can be animated controlling the animation of the components of an assembly.

A particular application for Armatures is the control and animation of Character Models. Figure 22.1 shows an **Armature** (blue bones) inside a character model. Each bone is linked to part of the surface mesh. The bones are animated to move which causes the surface mesh to move, posing the character.

The complete assembly of model, mesh, armature and controls is called a **Rig.**

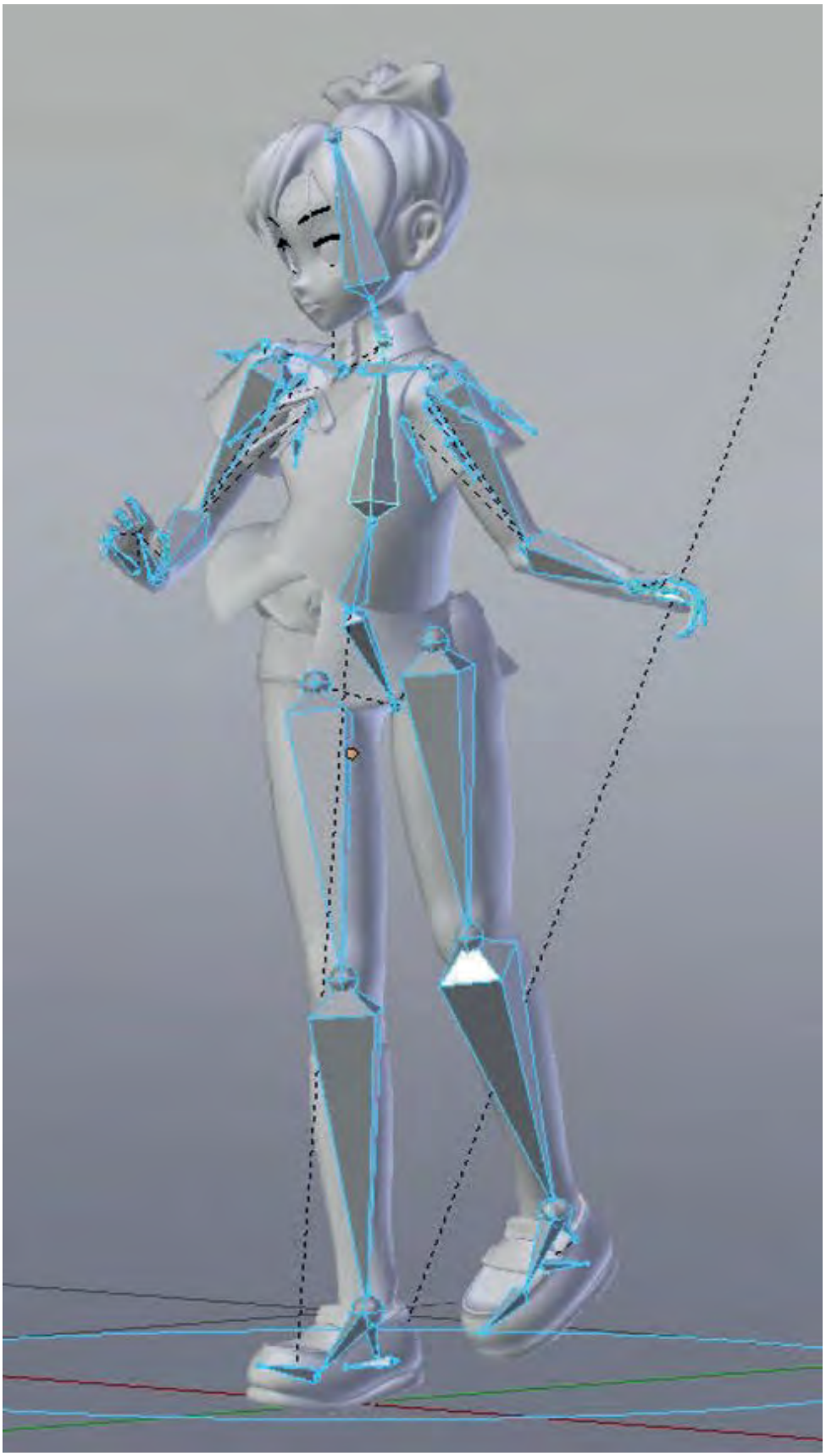


Figure 22.1

22.1 Single Bone Armature

Armatures can, and usually do, comprise multiple Bones as depicted in Figure 22.1 which is said to be an **Armature Rig** since the arrangement of Bones is rigged to fit the Character Model allowing the Model to be posed for Animation.

To understand how the Armature Rig is constructed, start with the default single **Bone Armature** (Figure 22.2). The single Bone is entered in the 3D Viewport Editor by clicking the **Add** button in the Header or pressing **Alt = A Key** on the keyboard and selecting **Armature** in the menu that displays.

The default **Single Bone Armature** display is **Type: Octahedral** (due to the object having eight surfaces): The Bone appears as two four-sided pyramids conjoined at the base with spheres at the apexes. For the purpose of the demonstration, the parts of the Armature will be named **Tip, Body,** and **Root** (Figure 22.2).

Note: Type: Octahedral is one of five display **Types** (explanation to follow).

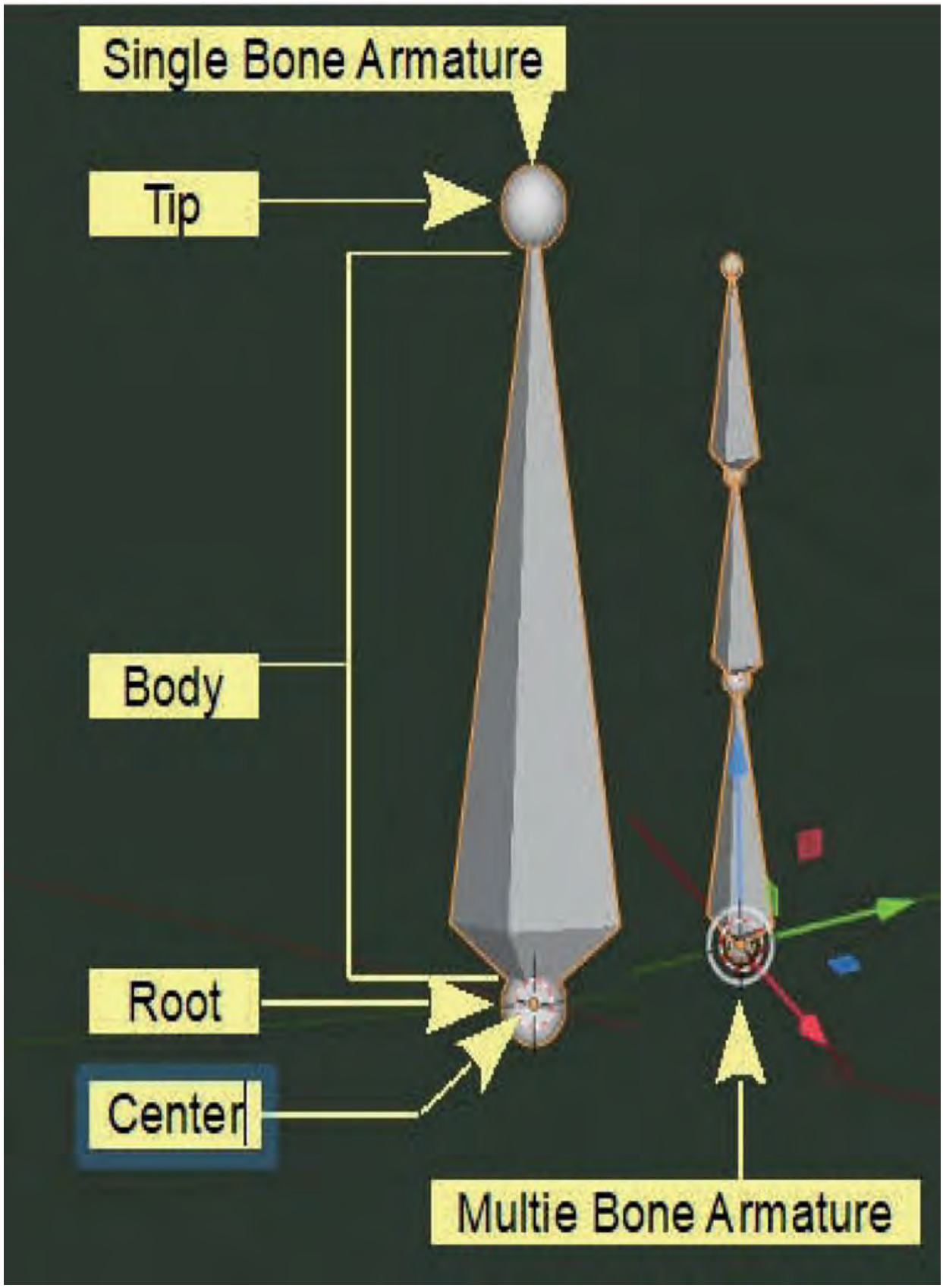

Single Bone Armature
Tip
Body
Root
Center
Multie Bone Armature

Figure 22.2

The Armature in Blender, it **is not** a Mesh Object. Its shape cannot be edited other than scaling it larger or smaller. It may be Rotated and Translated and It has a center like an Object. By default the Object Center is located at the center of the Root.

When the Manipulation Widget is used it locates at the Center.

When an Armature is entered in a Scene it displays in the 3D Viewport Editor and its name is automatically listed in the **Outliner Editor** (Figure 22.3).

Remember: Objects in a Blender Scene may be selected by clicking the Object in the 3D Viewport Editor or by clicking the Object's Name in the Outliner Editor. When dealing with Armatures and Armature Rigs comprising multiple Bones, this feature is particularly useful for selecting individual Bones.

The default Single Bone Armature is listed in the Outliner Editor with the name **Armature** (white lettering). The components of the Armature are: **Armature** and **Pose.** Under the Armature Component you see **Bone** which signifies that the Armature consists of a single Bone. These entries are the names of the **Datablocks** of information generating the display in the 3D Viewport Editor. The Pose component is the Datablock specific to the orientation.

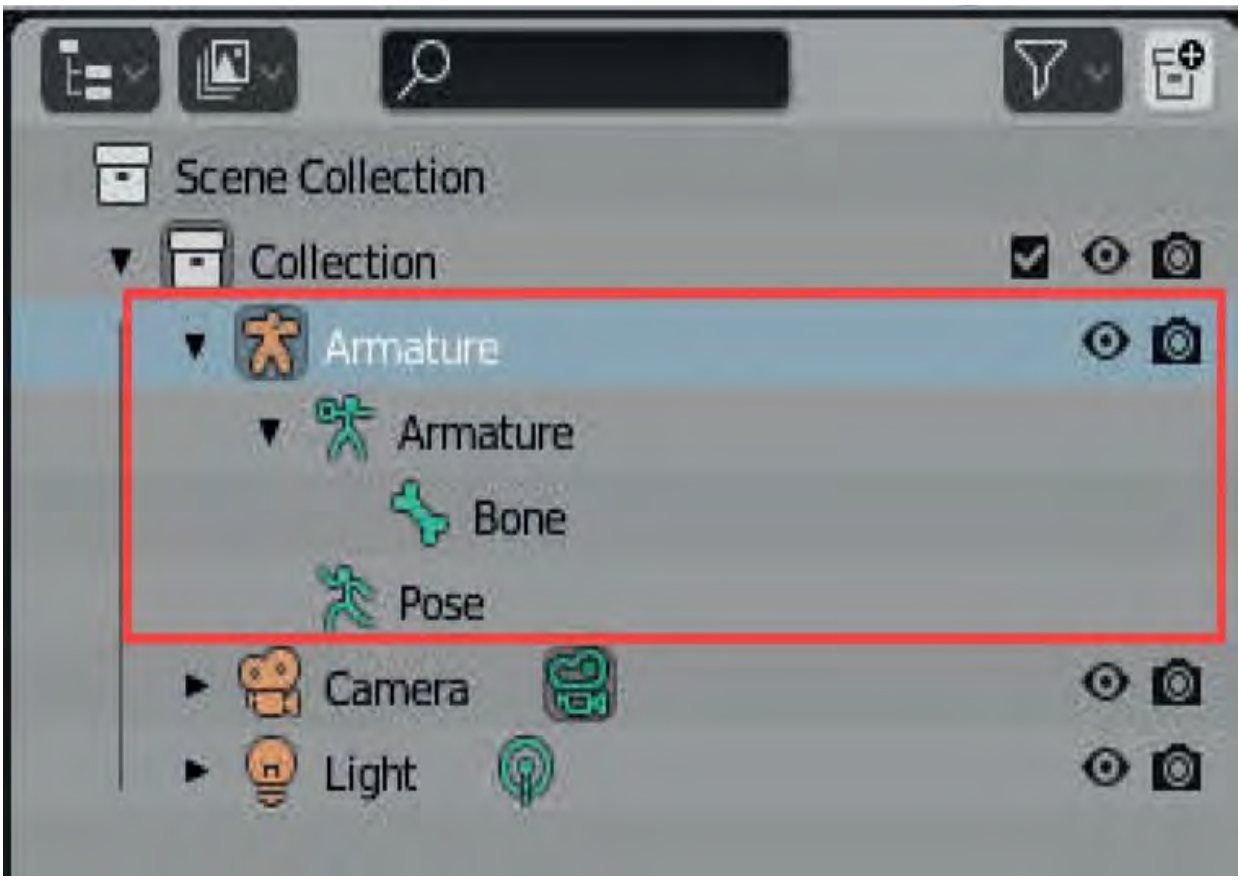


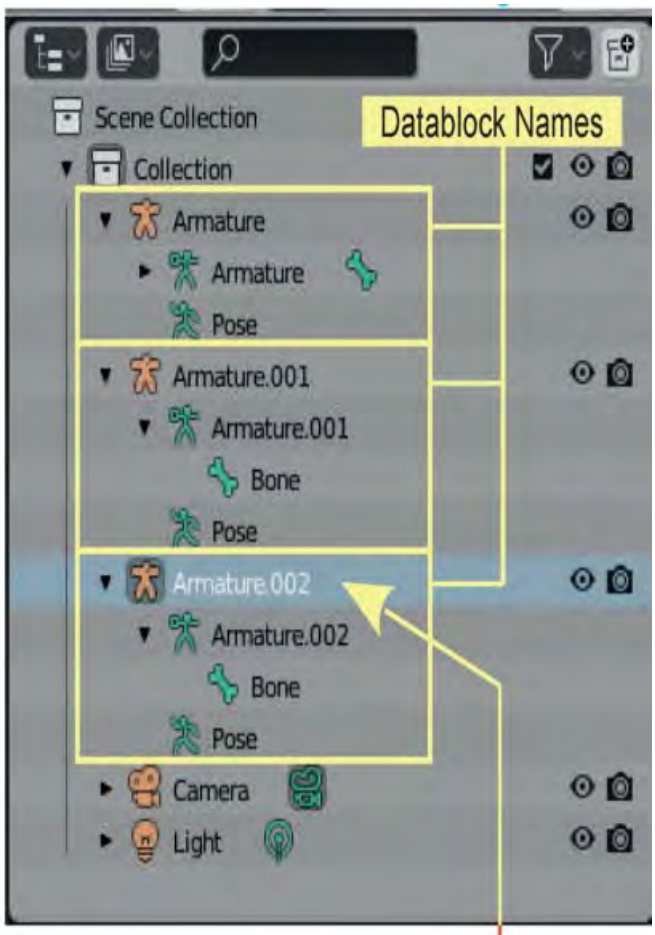
Figure 22.3

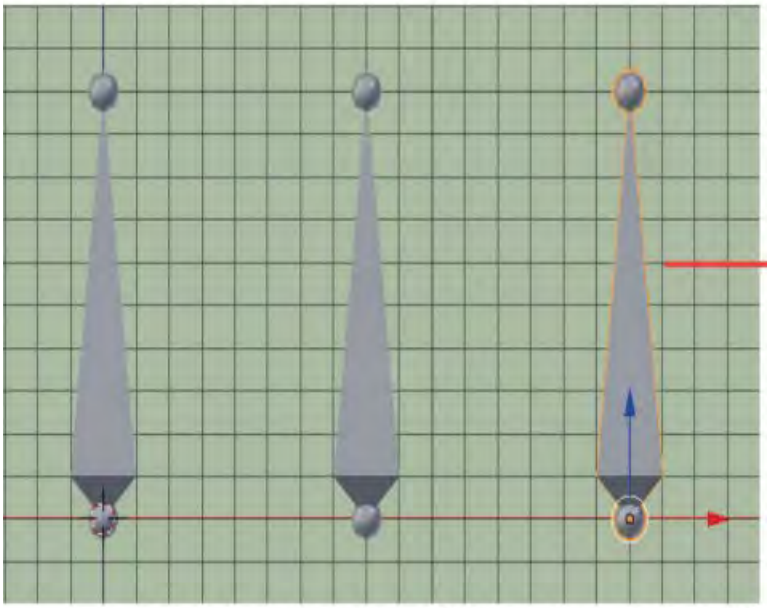
22.2 Adding Armatures

An Armature is added to the Scene from the **Add Menu** in the 3D Viewport Editor Header or by pressing **Shift + A Key.** It is located wherever you positioned the Editor Cursor (Reference Chapter 4 - 4.9), just like any other Object . You will see **Armature** listed in the **Outliner Editor** (Figure 22.4).

If you relocate the Cursor and repeat the process you add a second **Single Bone Armature.** Note: The new name, **Armature.001** in the **Outliner Editor** and the sub entries, **Armature.001** and **Pose** (Figure 22.4). If you select either Armature in the 3D Viewport Editor and press **Shift + D key** (Duplicate) and Translate (drag the mouse) you create a third single Bone Armature. The name in the **Outliner Editor** for this third Armature is **Armature.002.**

Note: All three Armatures are a single Bone and independent of each other.





Armature.002
Selected
(White Text)

Armature Armature.001 Armature.002

Figure 22.4
Figure 22.5

Note: Datablock Names for each Armature may be independent of the Armature Name.

Scene Collection
Collection
Armature
Armature
Bone
Bone.003
Bone.002
Bone.001
Bone.004
Bone.002_L
Bone.002_L.001
Bone.002_L.002
Bone.002_L.003
Bone.002_R
Bone.002_R.001
Bone.002_R.002
Bone.002_R.003
Pose
Camera
Light

Figure 22.6

As you see adding Armatures quickly accumulated names in the Outliner Editor. The list can become extensive when a Rig is complicated as is a Character Rig in the 3D Viewport Editor. Selecting components of a Rig in the 3D Viewport Editor can be challenging when the Rig is Posed, therefore, understanding the way components are listed in the Outliner and the automatic naming process is beneficial.

Figure 22.6 shows a listing of Bones making up an Armature Rig for a single finger of a human hand. If the hand is posed forming a fist, selecting part of a finger for fine tuning an animation is more easily done in the Outliner Editor.

The proviso is; you have to be organised. The automatic naming process is fine but renaming components to something meaningful can be an advantage.

22.3 Adding Bones

Adding Bones to an Armature is akin to extruding, subdividing or duplicating the Bones in the Armature.

Addoomg by Extrisopm (Figure 22.7)

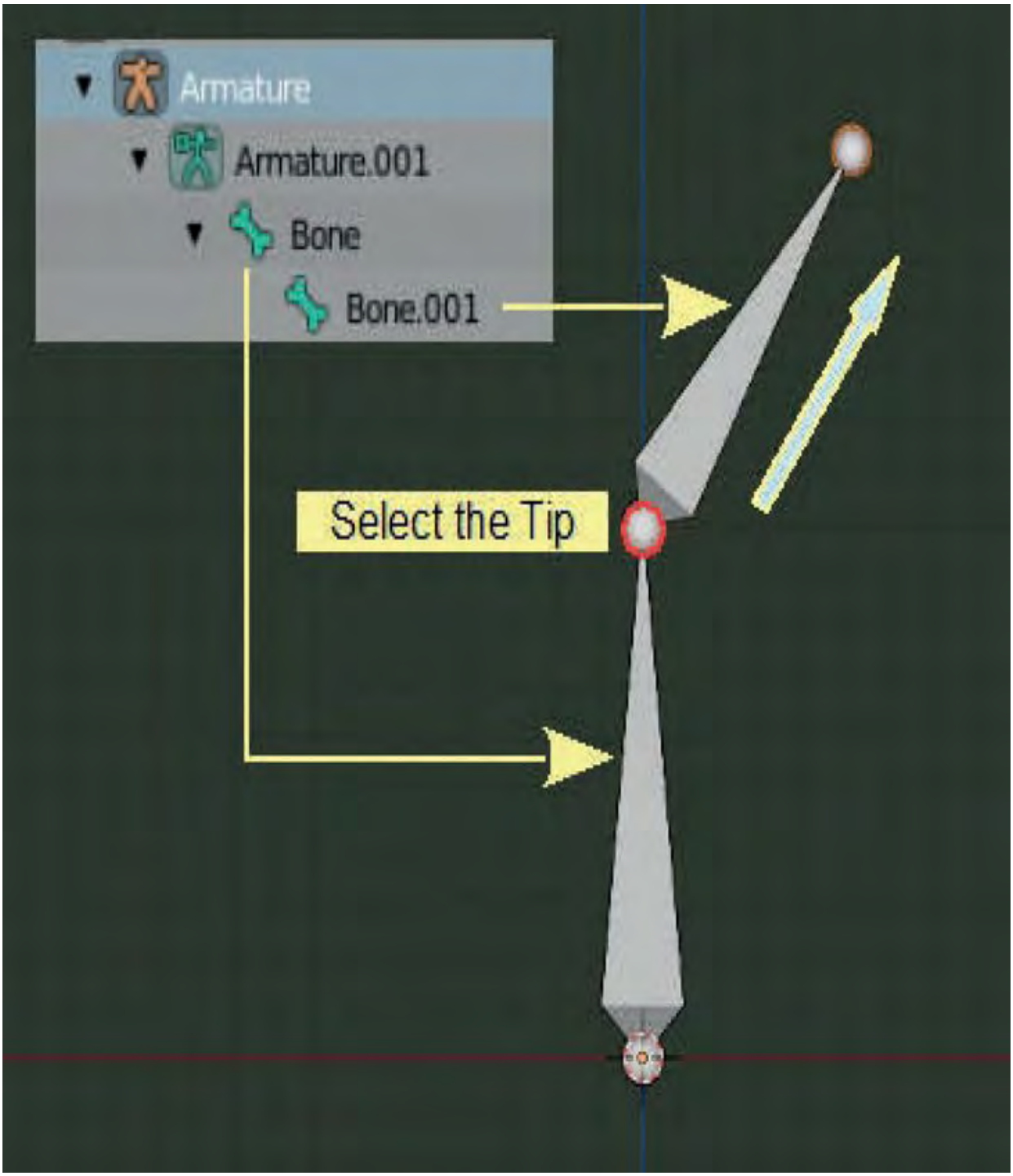
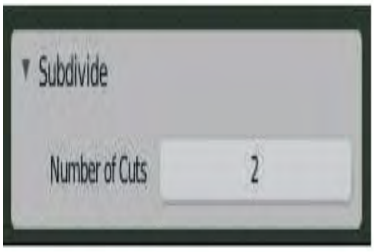


Figure 22.7

To add a Bone by Extrusion have a single Bone Armature selected in the 3D Viewport Editor **in Edit Mode.** When you switch from Object Mode to Edit Mode, the Tip of the Armature is selected by default. Press the **E Key**

and drag the Mouse, Extruding a new Bone. You may drag in any direction or confine the extrusion to an Axis.

Adding by Subdivision (Figure 22.8)



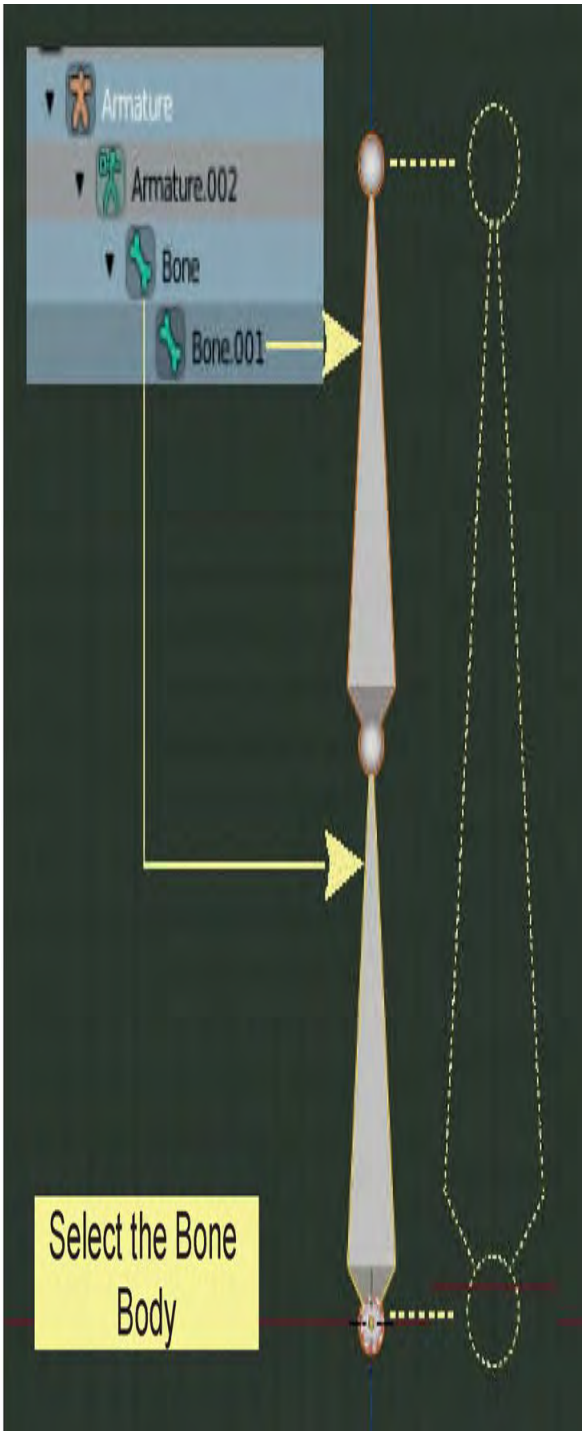


Figure 22.8

Have a single Bone Armature selected **in Edit Mode** (select the Bone Body). **Click RMB** in the 3D Viewport Editor and select **Subdivide** in the Menu that displays. The single Bone becomes Bone and Bone.001.

Note: The **Last Operator** (Subdivision Panel that displays at the lower left of the Screen.

Adding by Duplication (Figure 22.9)

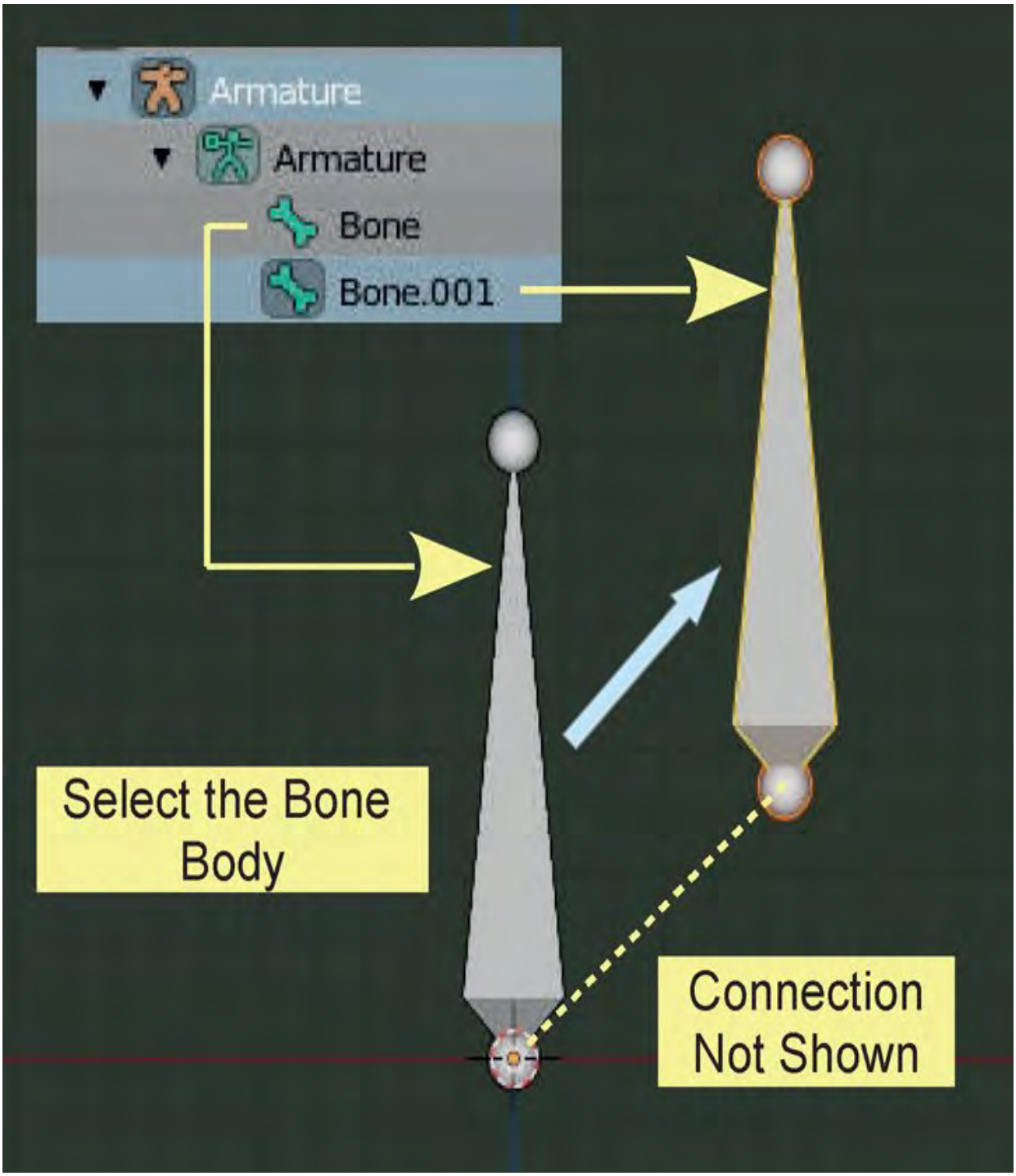


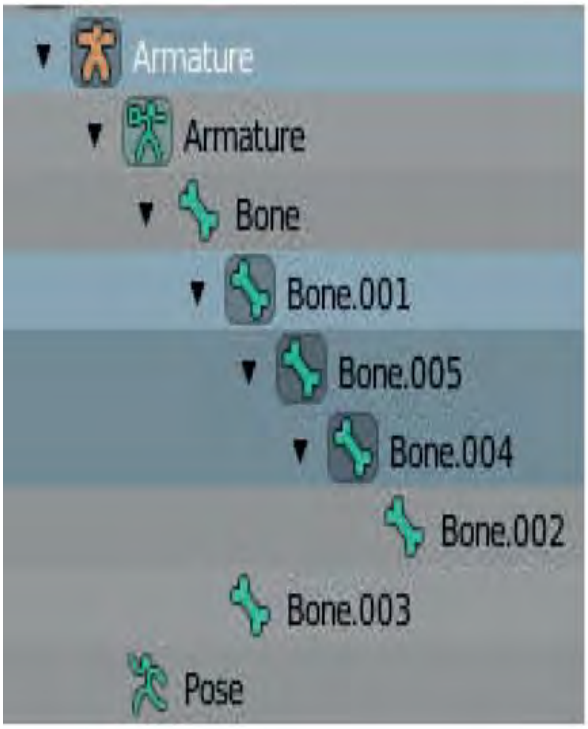
Figure 22.9

Have a single Bone Armature selected **in Edit Mode.** Press **Shift + D** and drag the Mouse to position the duplicated Bone (Bone.001).

Note: The new Bone (Bone.001) is a duplicate of the of the original Bone not a New Bone and although not shown, it is connected to the original.

In **Object Mode,** when the original Bone is Translated, Bone.001 follows the original.

Extruding and Subdividing


Armature
Armature
Bone
Bone.001
Bone.005
Bone.004
Bone.002
Bone.003
Pose

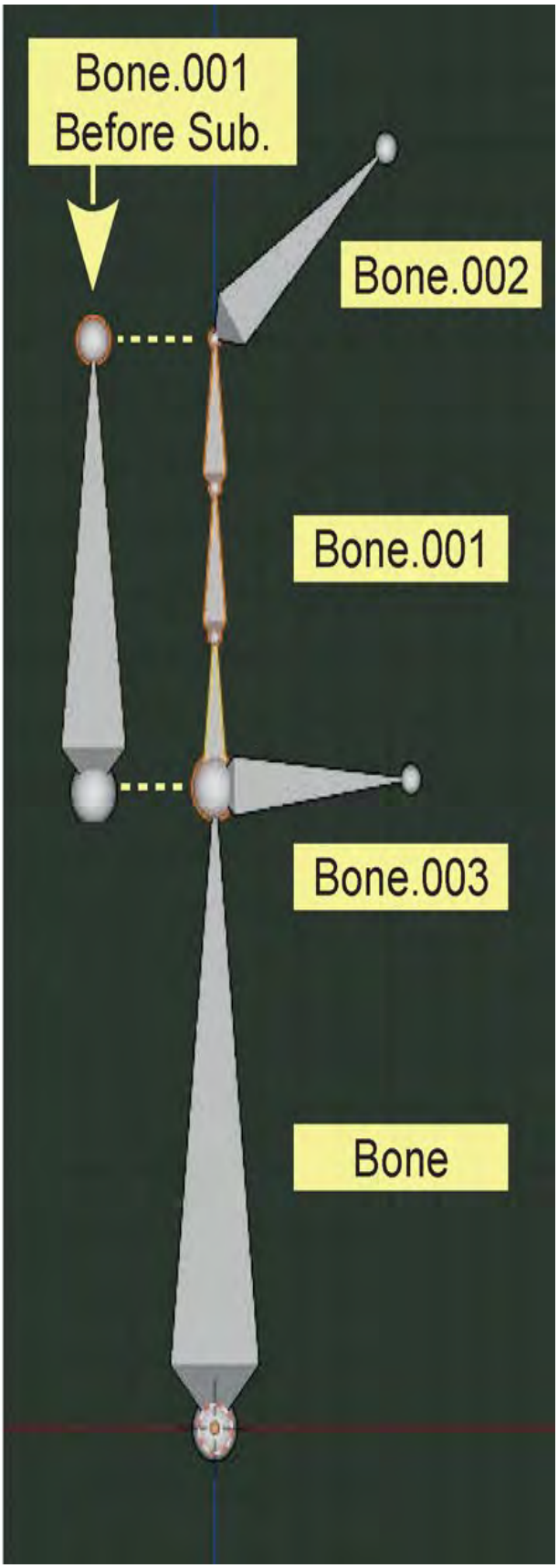

Bone.001
Before Sub.
Bone.002
Bone.001
Bone.003
Bone

Figure 22.10

Bones may be Extruded from the Tip of any Bone, in any direction, when in Edit Mode. Any Bone may be Subdivided into any number of Bones.

In Figure 22.10 Bone.001 has been Extruded from the Tip of Bone. Bone.002 has been Extruded from the Tip of Bone.001. Bone.003 has been extruded from the Tip of the original Bone. Bone.001 has been Subdivided twice creating Bone.004 and Bone.005.

The Bone Tree in the Outliner Editor has suddenly become a little complicated and without names being displayed in the 3D Viewport Editor it is becoming difficult to determine which Bone is which.

You may display the Bone Names (Numbers) in the 3D Viewport Editor by checking Names in the Properties Editor, Object Data Properties, Viewport Display Tab.

With the 3D Viewport Editor in Edit Mode, clicking on a Bone name (number) in the Outliner selects the Bone in the 3D Viewport Editor. This procedure is essential for Posing the Bones for animation.

22.4 The Armature Rig

In adding Bones to the Armature creating the arrangement shown in Figure 22.10 you have created an **Armature Rig.** In practice the Rig would be constructed to fit within a Model where the individual Bones are linked to parts of the Model such that when a Bone is moved the linked part of the Model follows the movement. This arrangement is being described in general terms since Armatures are used to control and animate not only characters but model you may wish to construct. For example, machine parts.

The individual Bones forming the Armature Rig are connected in a hierarchy such that when one Bone moves, others follow, depending on where they are positioned in the hierarchy.

The way in which one Bone follows another is termed a **Child Parent Relationship** (reference 22.6).

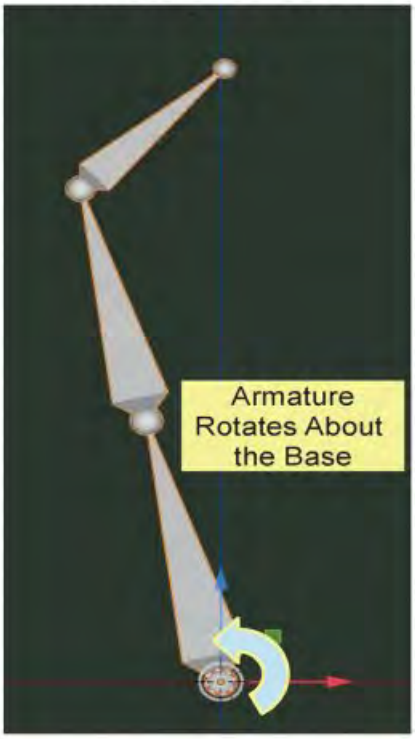
Before discovering how the individual Bones in a Rig interact in this relationship you should be aware of the three Modes in construction when considering Armatures.

The three Modes are: Object Mode, Edit Mode and Pose Mode.

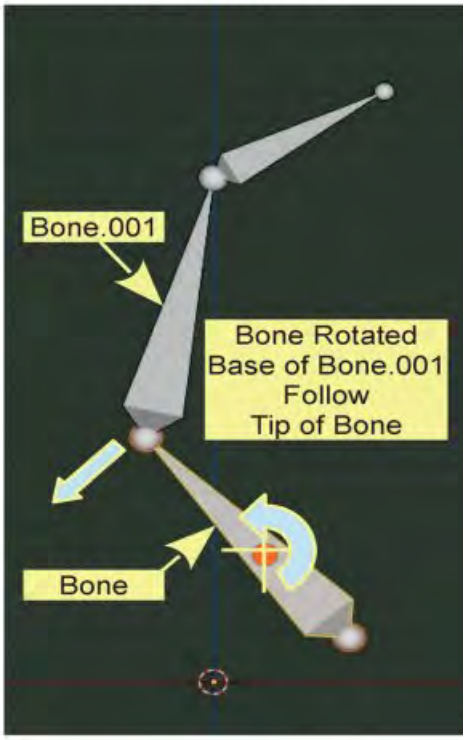
Translating, Rotating and Scaling the Armature and its components (the Bones) perform differently in each Mode.

Rotating the Armature and Bones

Object Mode
Armature Selected



Edit Mode
Bone Selected



Pose Mode
Bone Selected

Pose Mode
Bone.001 Selected

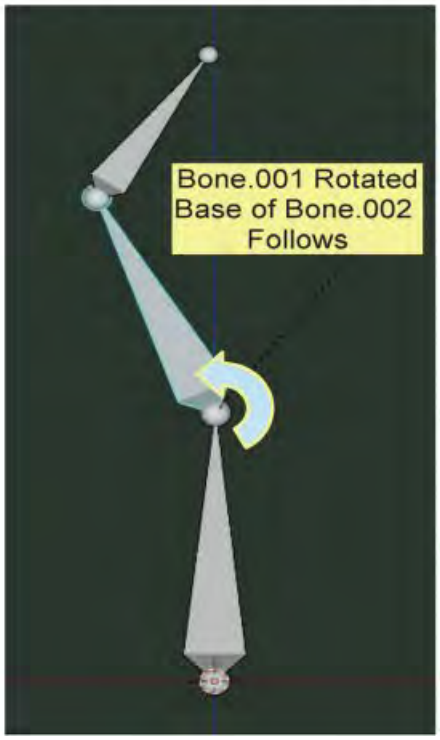


Figure 22.11

22.5 Multiple Armatures

In the preceding sections the discussion has shown how to add Bones to an Armature creating an Armature Rig. The process starts by adding a Single Bone Armature to a Scene (reference Figures 22.2 and 22.10) then adding Bones to the Armature creating a Rig. There will be occasions where you will have more than one Armature in a Scene

The Outliner Editor provides a graphical display of the Armatures and Bones in a Scene. **In Edit Mode** you can select a Bone in an Armature by clicking on its name in the Outliner Editor.

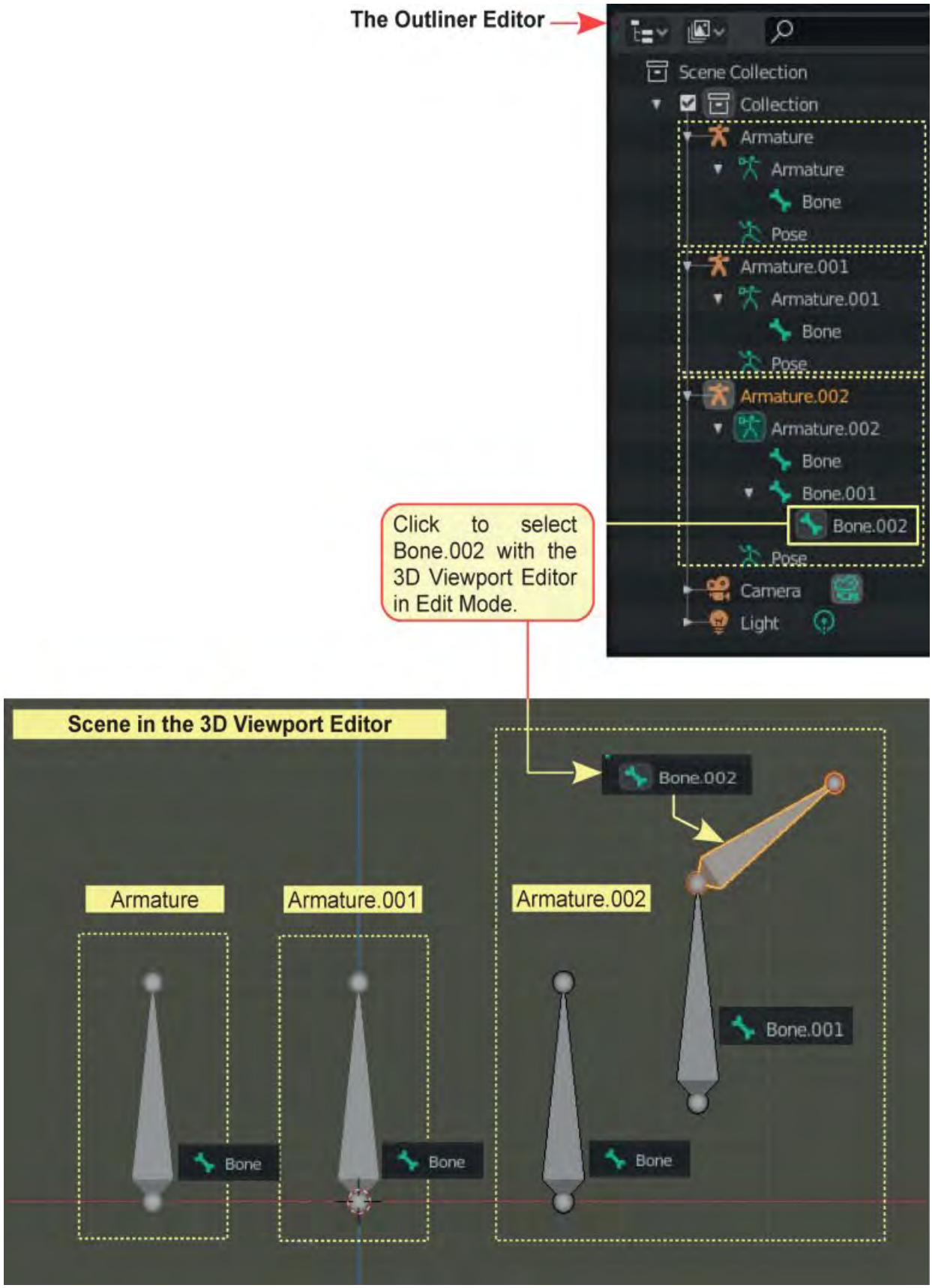
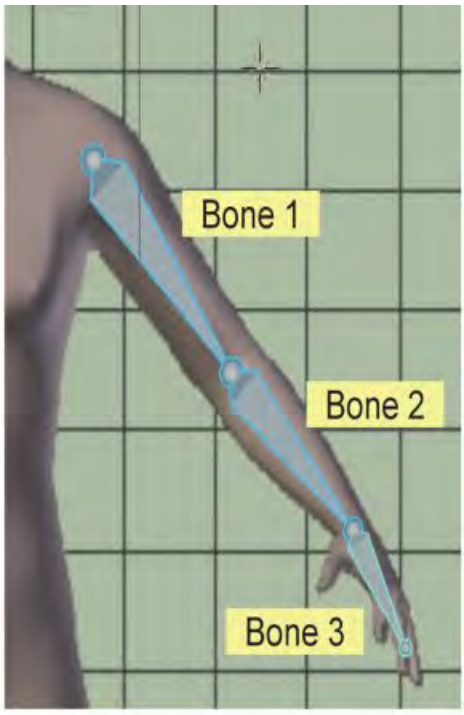

The Outliner Editor
Scene Collection
Collection
Armature
Armature
Bone
Pose
Armature.001
Armature.001
Bone
Pose
Armature.002
Armature.002
Bone
Bone.001
Bone.002
Pose
Camera
Light
Click to select Bone.002 with the 3D Viewport Editor in Edit Mode.
Scene in the 3D Viewport Editor
Bone.002
Armature
Armature.001
Armature.002
Bone.001
Bone
Bone
Bone

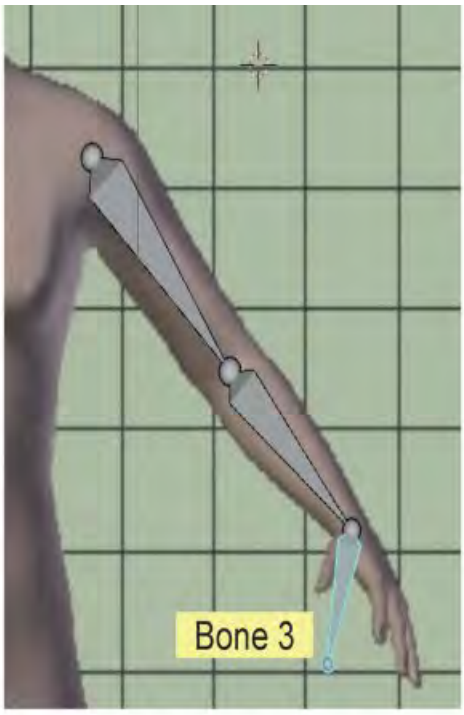
Figure 22.12
Figure 22.13

22.6 Child Parent Relationship

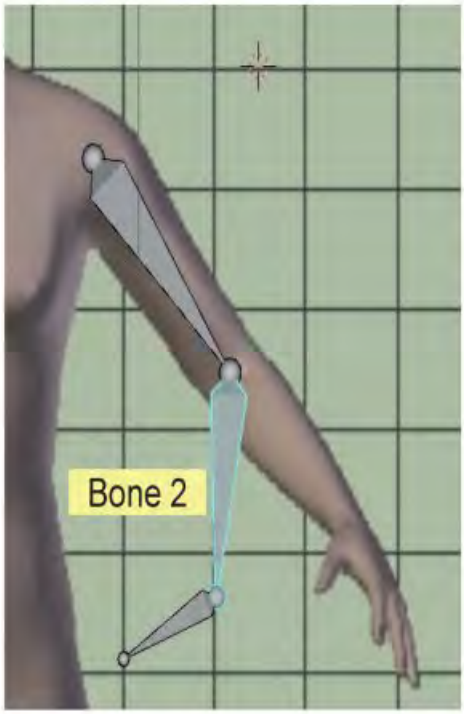
The Bones in an Armature are connected in a **Child Parent Relationship.** When a Bone (Bone 2) is extruded from the tip of an Armature (Bone 1) it automatically becomes the Child of Bone 1. Extruding another Bone (Bone 3) from the tip of Bone 2 makes Bone 3 the Child of Bone 2. Being a Child means that the Bone follows its parent.



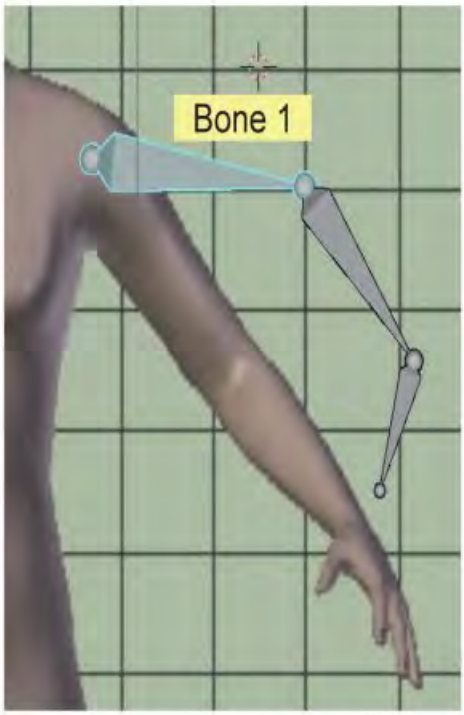
Bone 2 Extruded from Tip of Bone 1.
Bone 3 Extruded from Tip of Bone 2.



Bone 3 Rotates independently
but is fixed to Bone 2.



Bone 2 Rotated Bone 3 Follows.
Bone 3 is the Child of Bone 2.



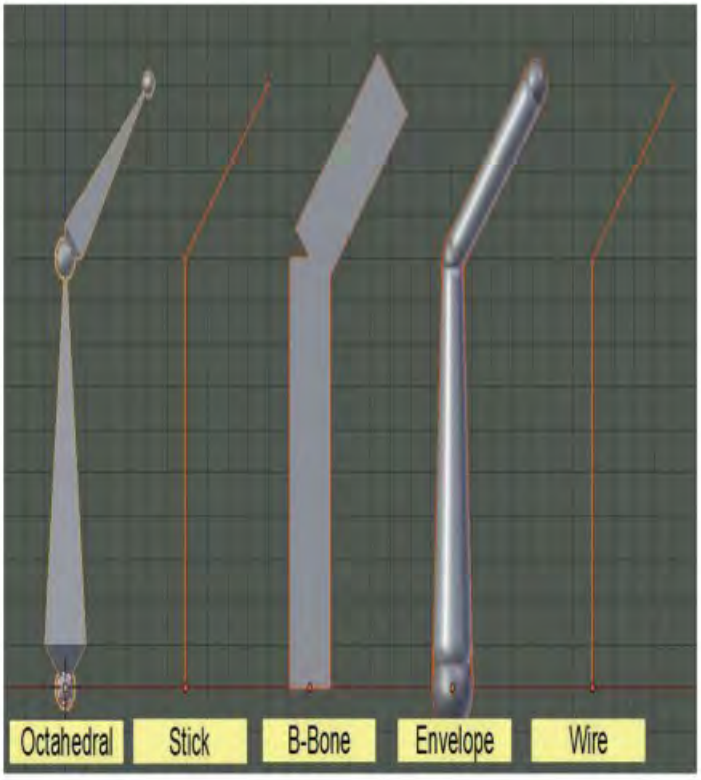
Bone 1 Rotated Bones 2 and 3 Follow.
Bone 2 is the Child of Bone 1.

Figure 22.14 Pose Mode

Note: With the Bones linked to the mesh forming the arm, the Mesh will follow the Bones.

22.7 Armature Display Types

The default Armature display type is **Octahedral.** There are four alternative display types



With the Armature Bone selected, see the **Properties Editor, Data buttons, Viewport Display Tab.**

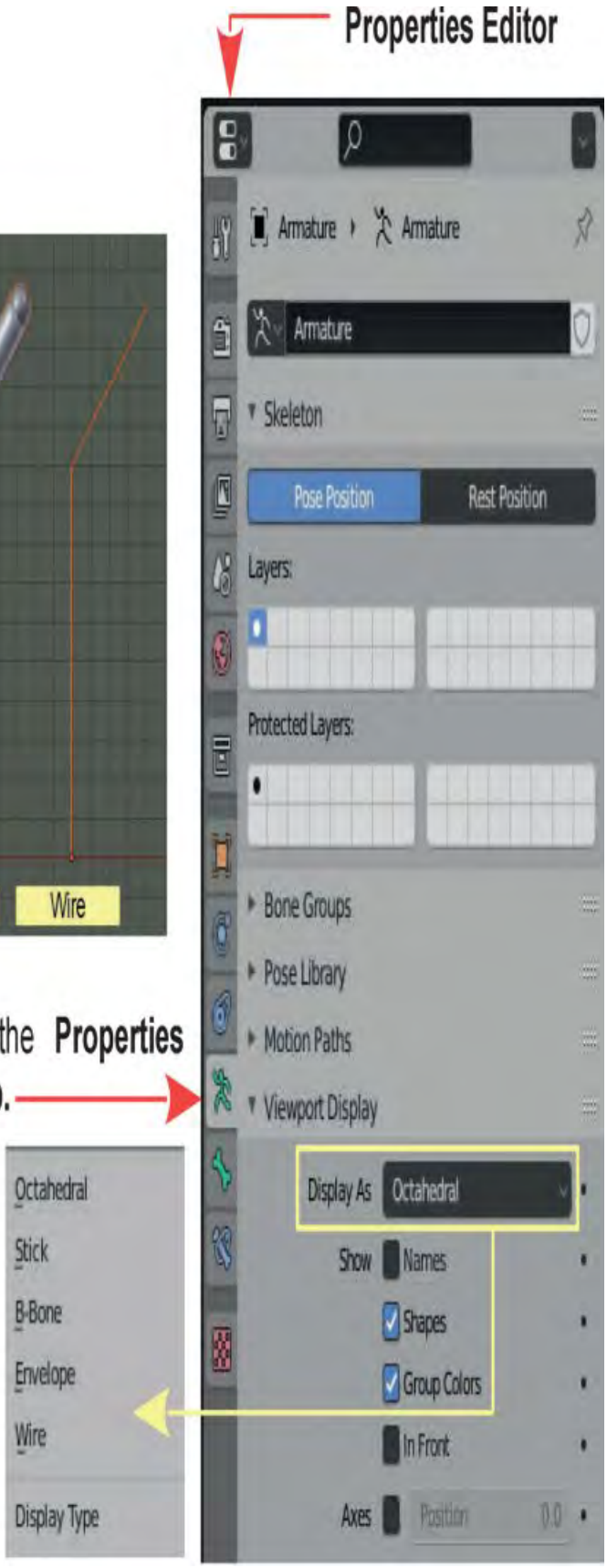


Figure 22.15

Which display type is used depends on what you will do with the Armature. The different uses will not be explained at this time but since the basic function of an Armature is to deform a mesh Object, you need to understand how this occurs.

22.8 Building an Armature Rig

In adding Bones to an Armature, as you did in Figure 22.9, you created a Multi-Bone Armature which is essentially an Armature Rig.

Adding Bones to an Armature and creating an Armature Rig is primarily for posing or animating a Character Model. In Figure 20.1 (at the beginning of this chapter) you see an Armature Rig inside a Model. There are pre-assembled Rigs which may be used but you should understand how they are created. Understanding will allow you to create Rigs for any application. Rigs are employed for many characters such as strange creatures and weird robots, not only human figures. They are also used when animating machine parts.

To demonstrate how to create a Character Rig for a model, a model of a human figure will be used which has been generated in the **Make Human** program.

Figure 22.16

Make Human is a **free Open Source** human character modeling program. You can import a model from the program into a Blender file. Importing Objects is discussed in Chapter 3 -3.13.

Figure 22.17 shows the imported model with the Multi-Bone Armature fully constructed. The Armature is shown here so you can see what you are aiming for in the exercise. Figure 22.18 shows the Armature moved to one side as a reference for construction.

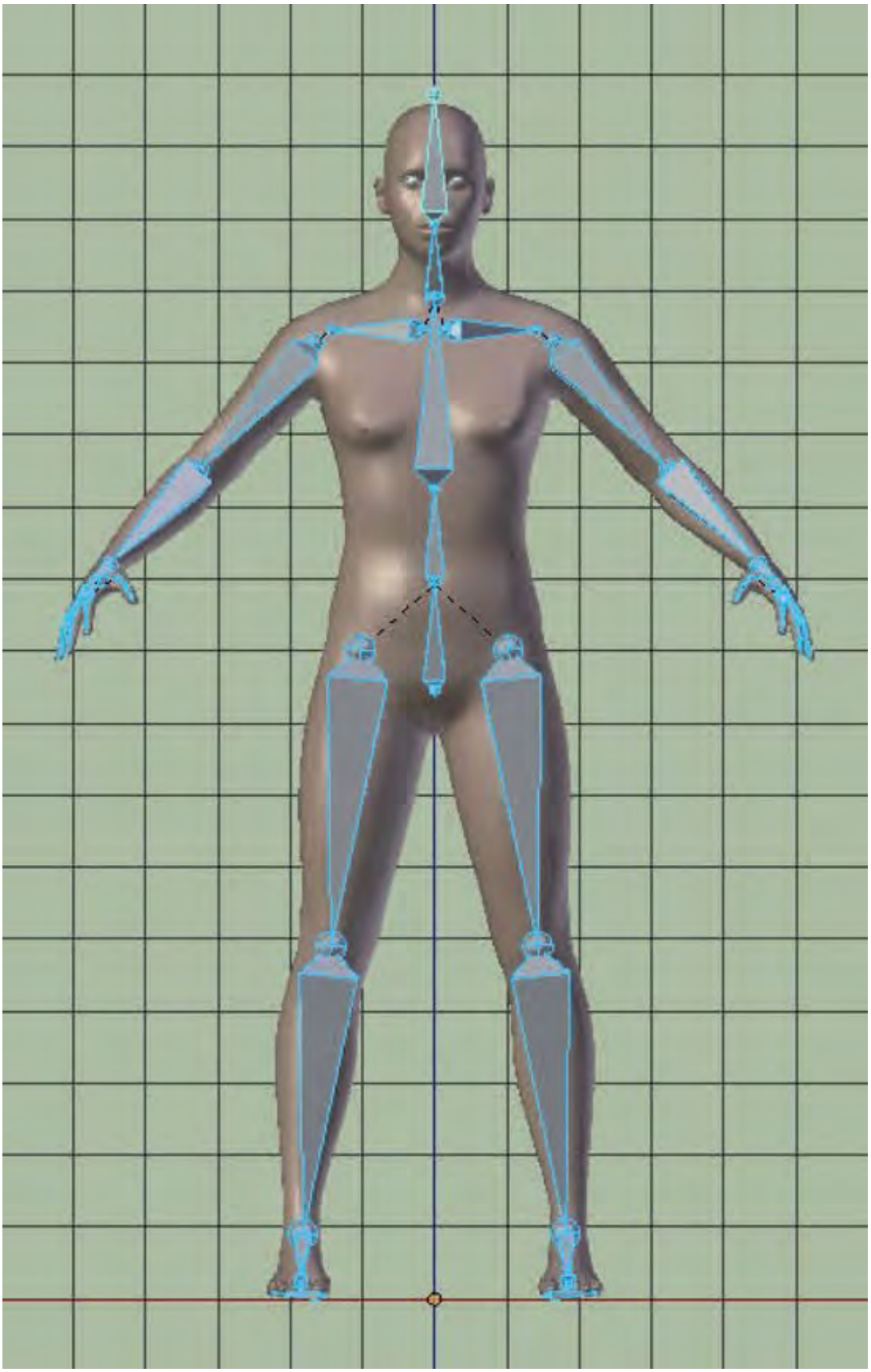


Figure 22.17

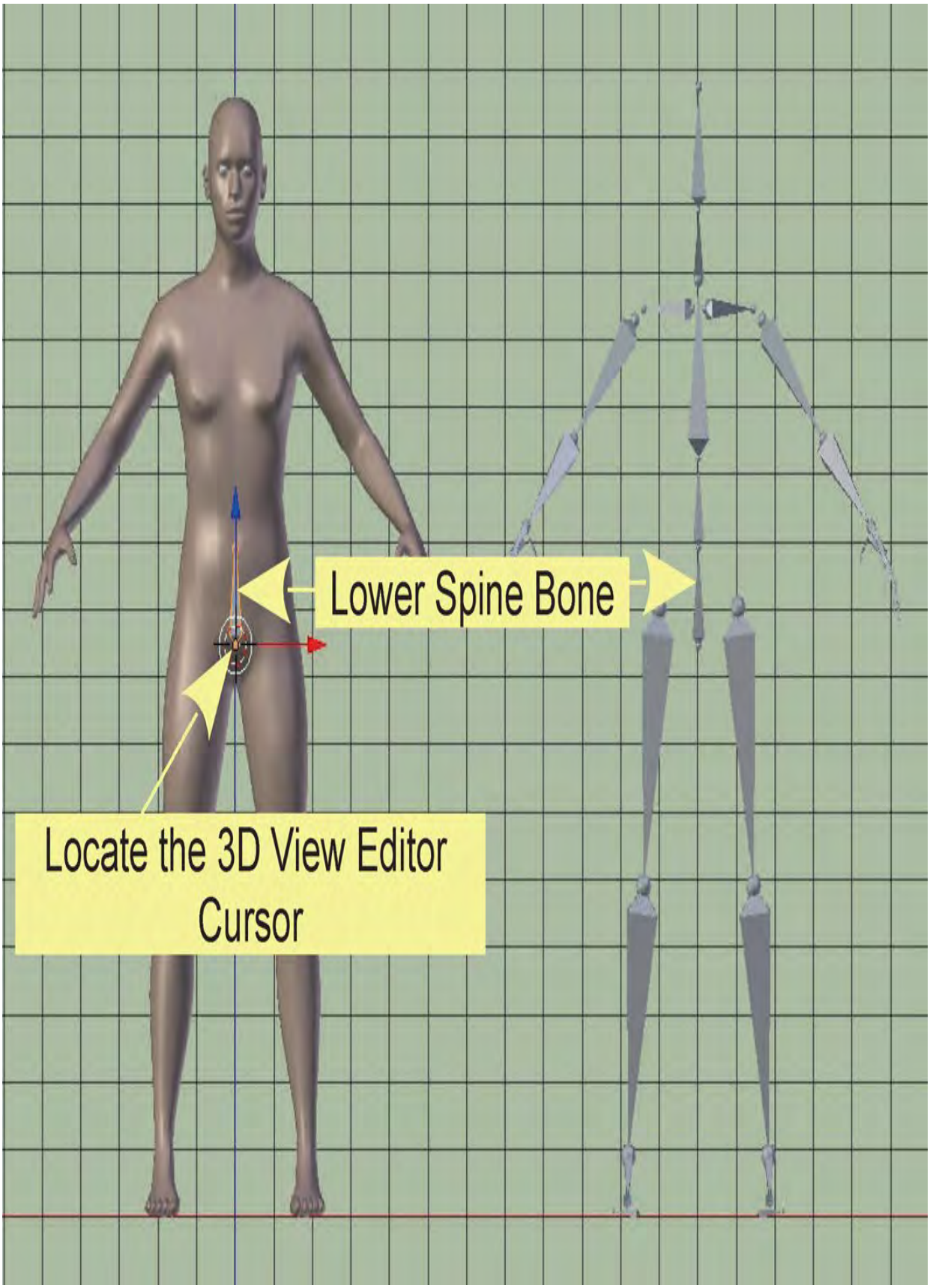

Lower Spine Bone
Locate the 3D View Editor Cursor

Figure 22.18

When constructing the Armature think of it as a human skeleton.

Obtaining the Model

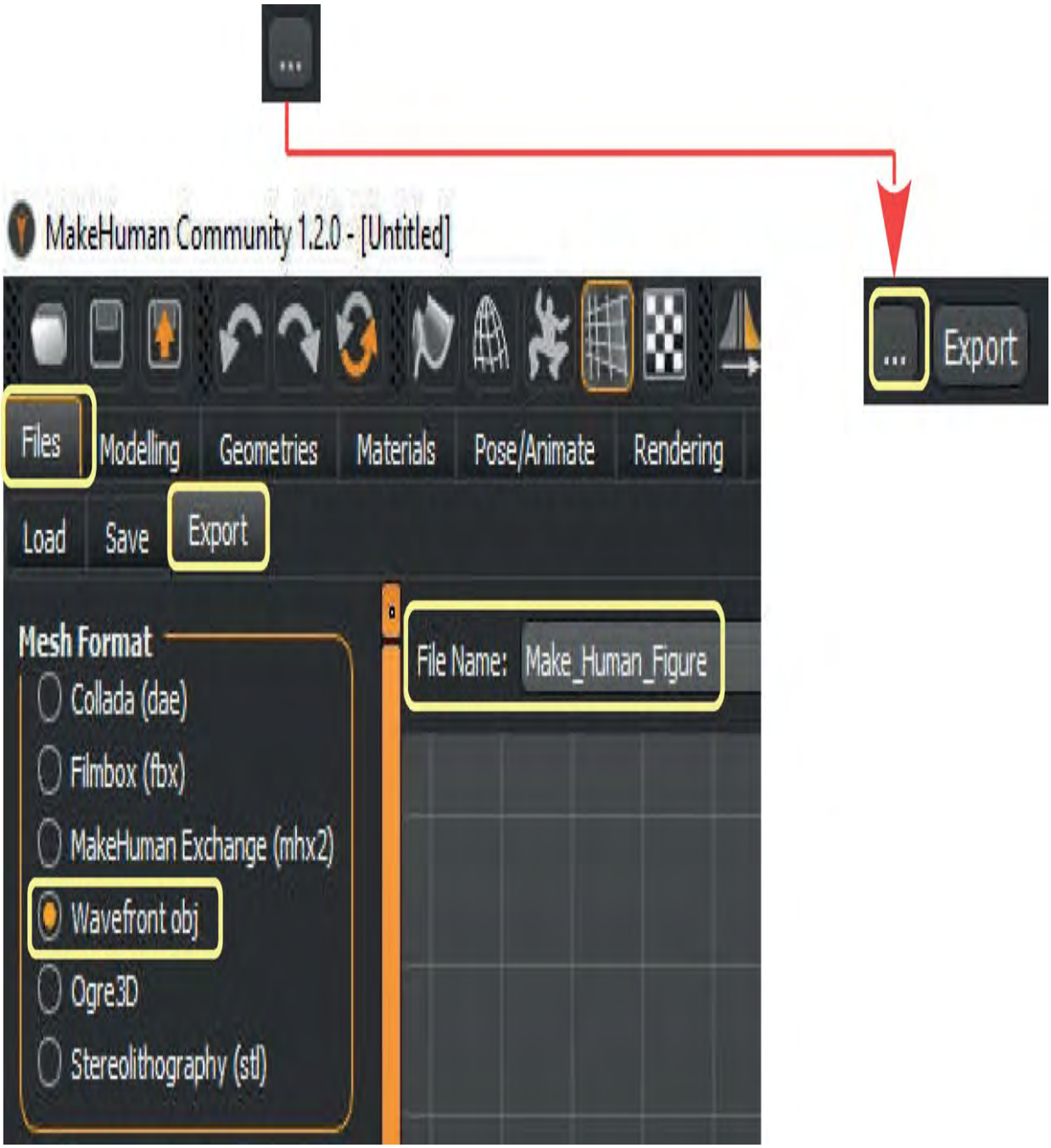


Figure 22.19

In Blender, select File – Import – Wavefront (.obj). Navigate to the folder where you saved the make Human file then click **Import OBJ.**

Note: the Model will be way too big. Scale down.

Rig Construction

To start the construction the Rig first Bone to be placed is the **Lower Spine Bone.** Locate the **3D Viewport Editor Cursor** where you want to place the Base of the Bone. Press **Shift + A key** and select **Add – Armature.** The single Bone Armature is entered in the Scene but it may be way too big or too small for the model. Scale to fit.

Note: When the Armature is scaled, depending on its location, it may disappear inside the model (Figure 22.21).

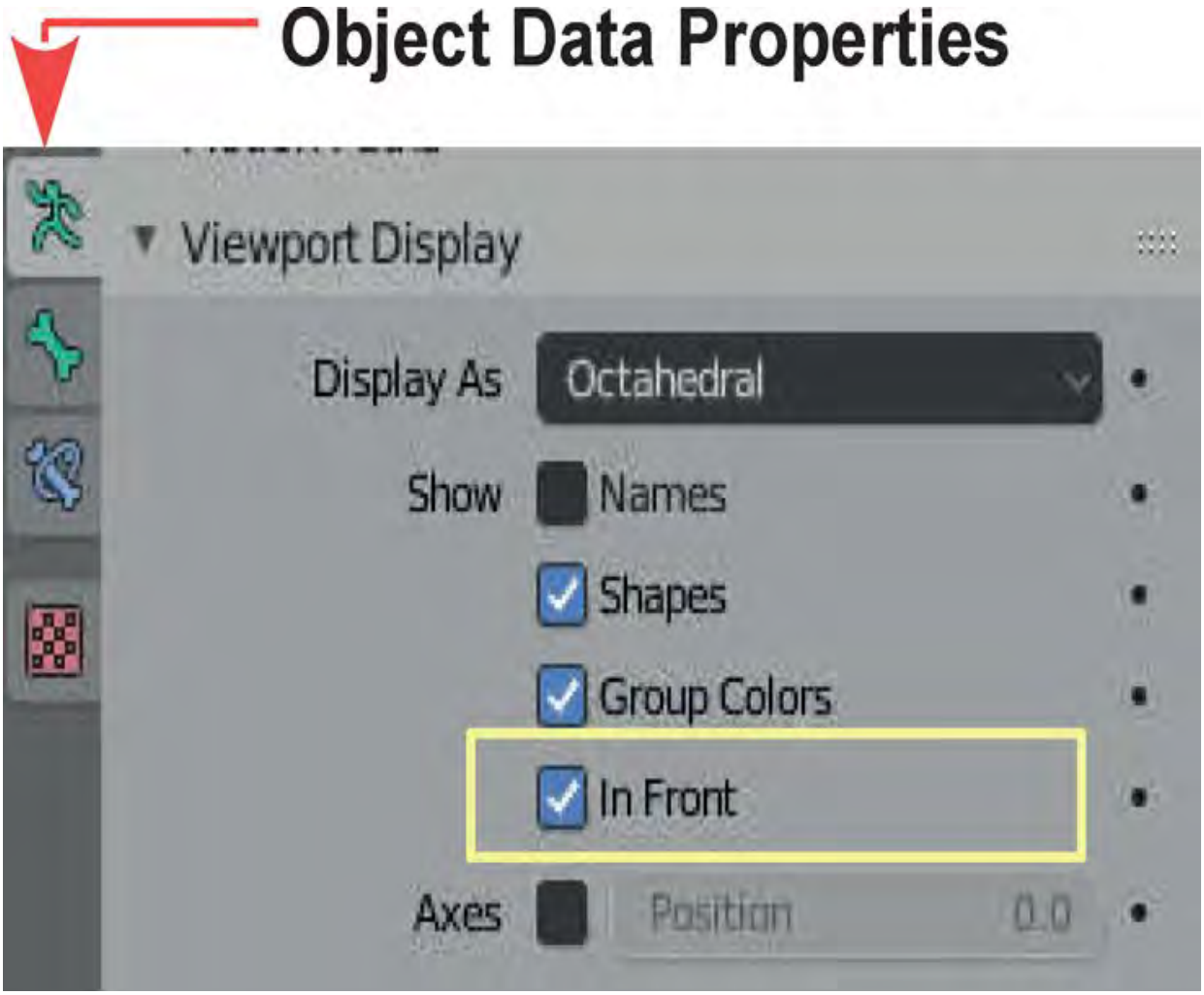


Figure 22.20

To see the Bone inside, check **In Front** in the **Viewport Display tab** in the **Properties Editor, Object Data Properties** (Figure 22.20). The bone must be selected in the 3D Viewport Editor.

With **In Front** checked you will see the Bone (Figure 22.22).

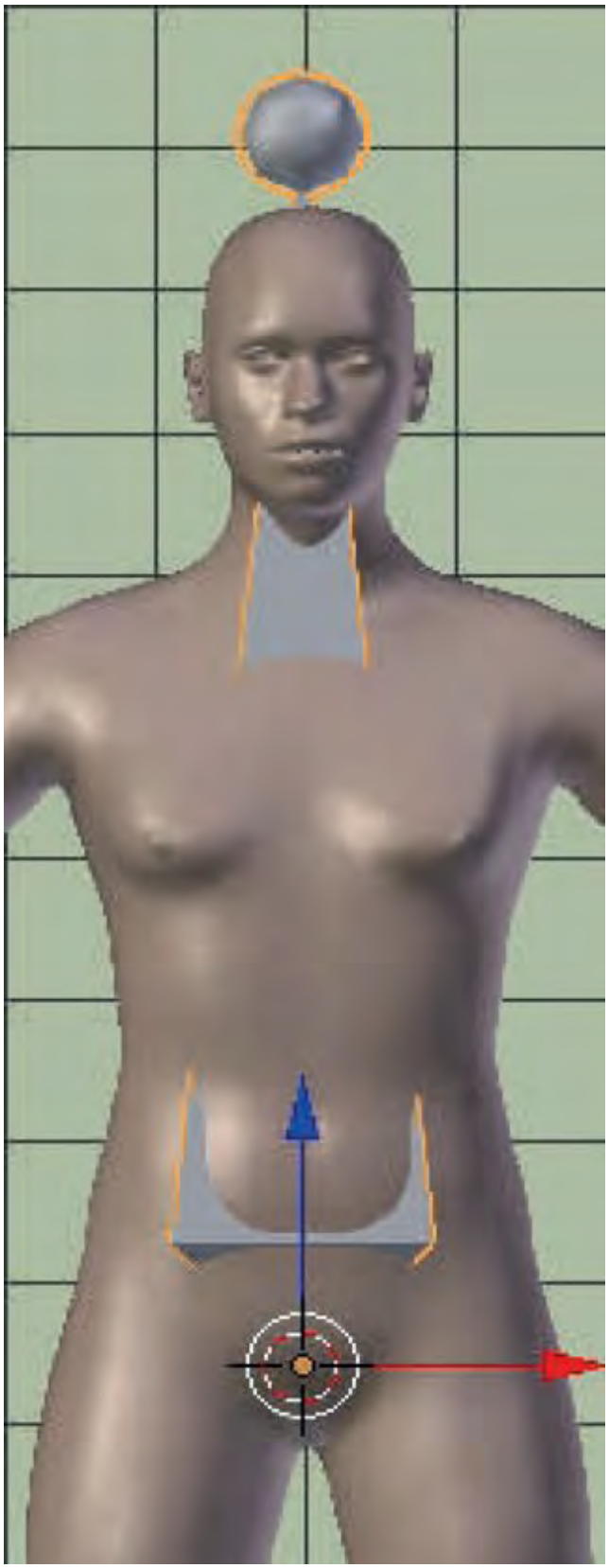


Figure 22.21

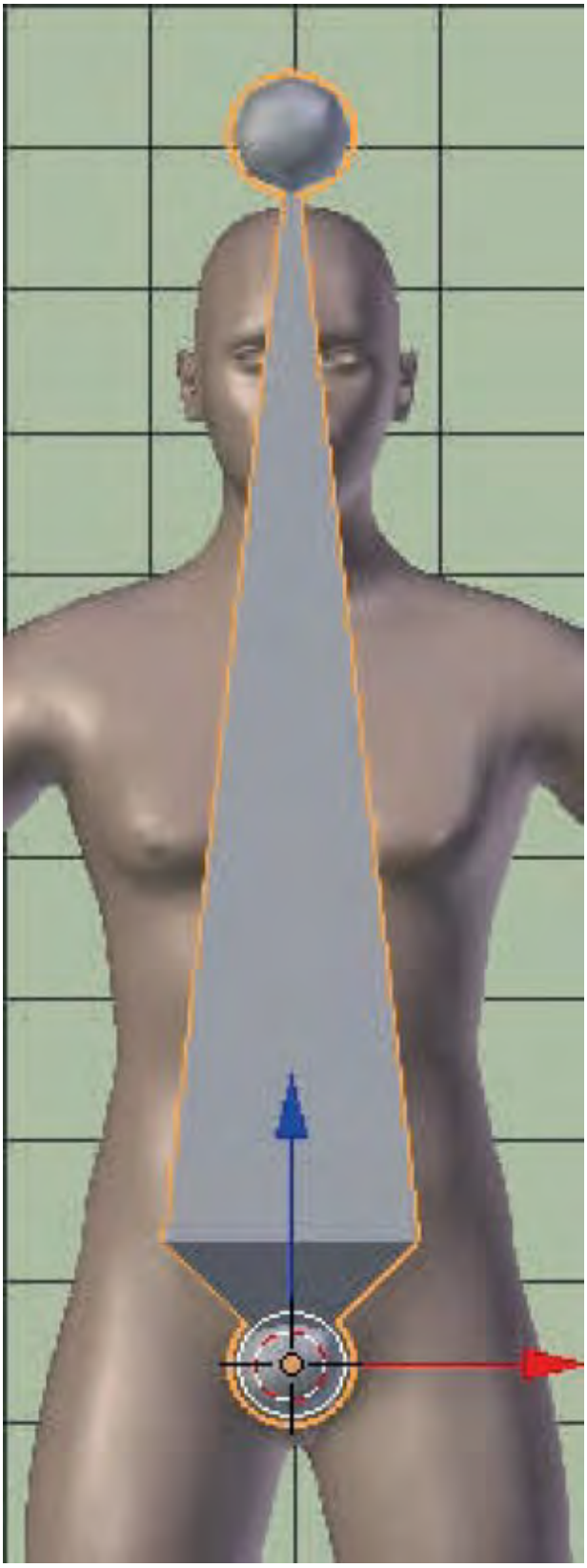


Figure 22.22

Creating the Spine

To create a **Spine** from a single Bone, have the bone selected in the 3D View Editor and in Edit Mode drag the Tip up above the top of the model's head (Figure 22.22). In Edit Mode, RMB click on the Body of the Bone then select **Subdivide.** In the **Last Operator Panel Subdivide** panel that displays, increase the **Number of Cuts** to produce multiple Bones (Figure 22.23).

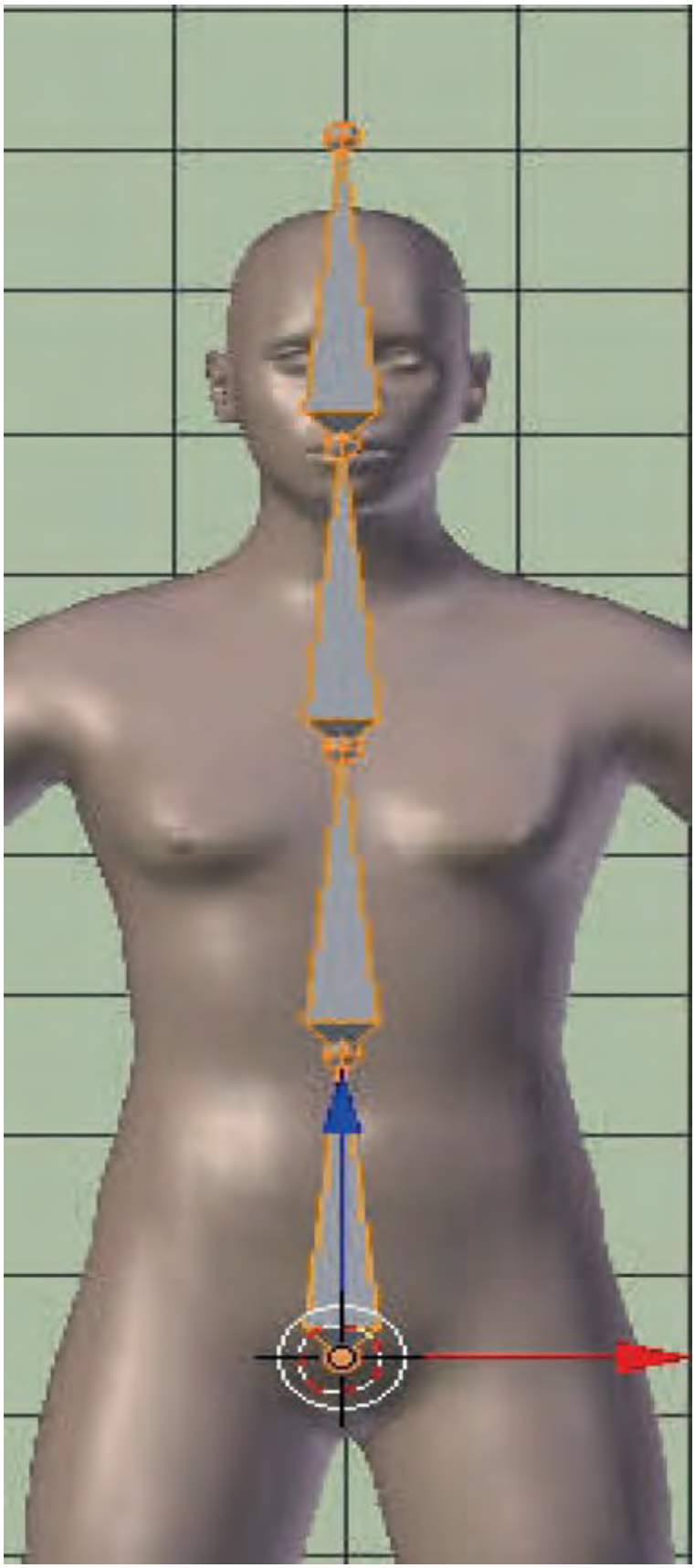


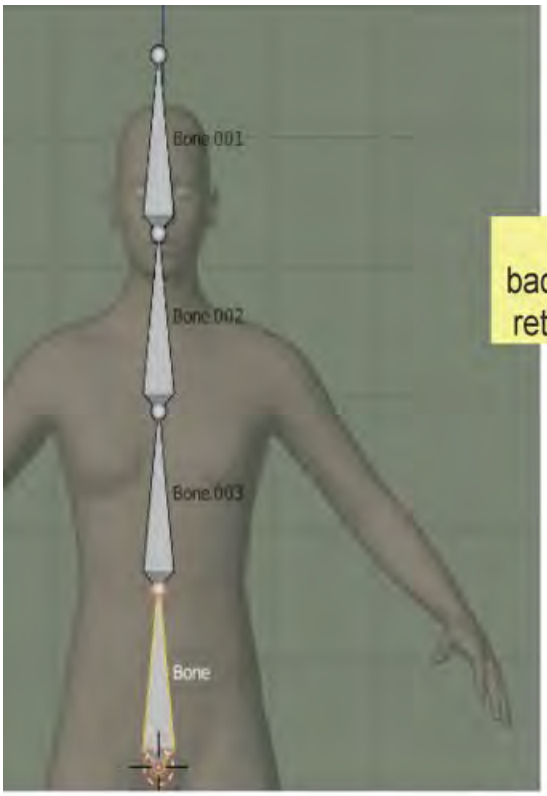
Figure 22.23

Remain in Edit Mode, deselect, then select individual Tips and Bases of the bones and Translate to fit the model. You may also select individual Bone bodies and Scale and Translate.

Note: In this instance the Spine has been created by Subdivision but you may create the Spine by selecting the Tip of the first single Bone, then extruding new Bones.

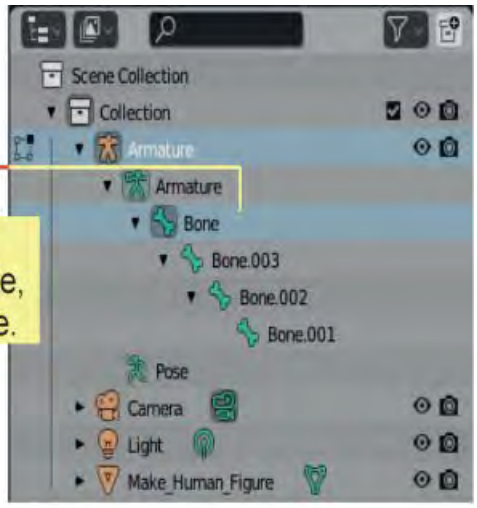
Naming the Bones

At this point it is good practice to begin naming the components of your Rig. With the Bones created by Subdivision or Extrusion names are automatically applied in the Outliner Editor and may be displayed in the 3D Viewport Editor by checking **Show Names** in the Properties Editor, Object Data Properties, Viewport Display Tab. Automatic names are fine but something more meaningful will be an advantage when the Rig is developed.

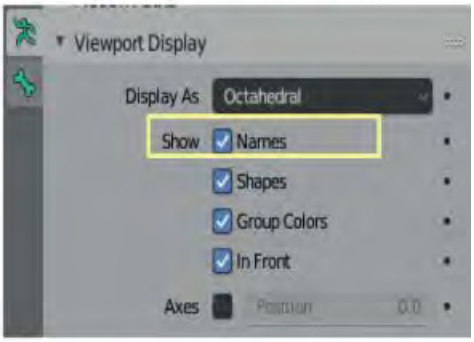


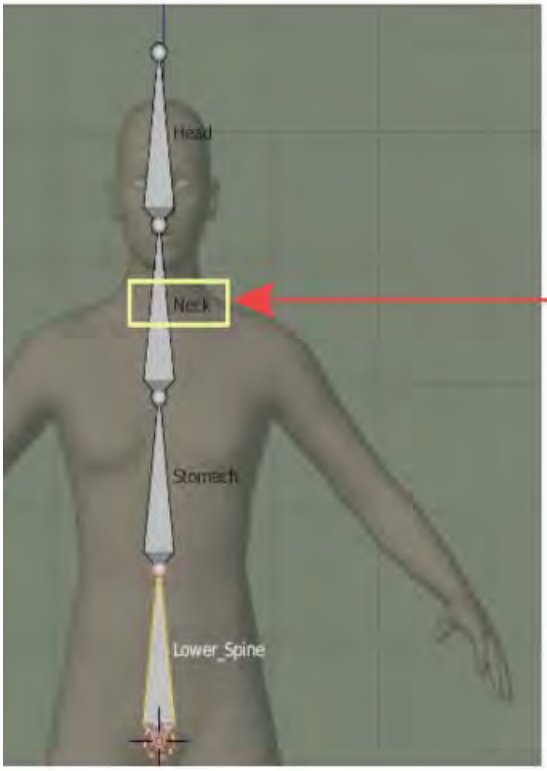
Automatic Names

Double click, backspace or delete, retype a new name.

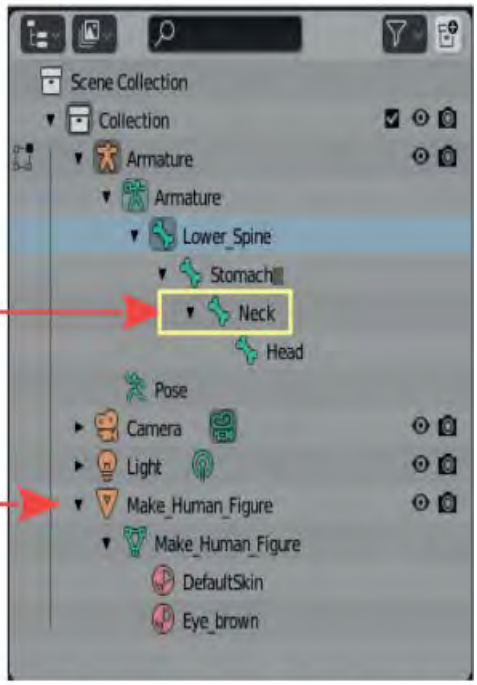


Show Names





Names Renamed



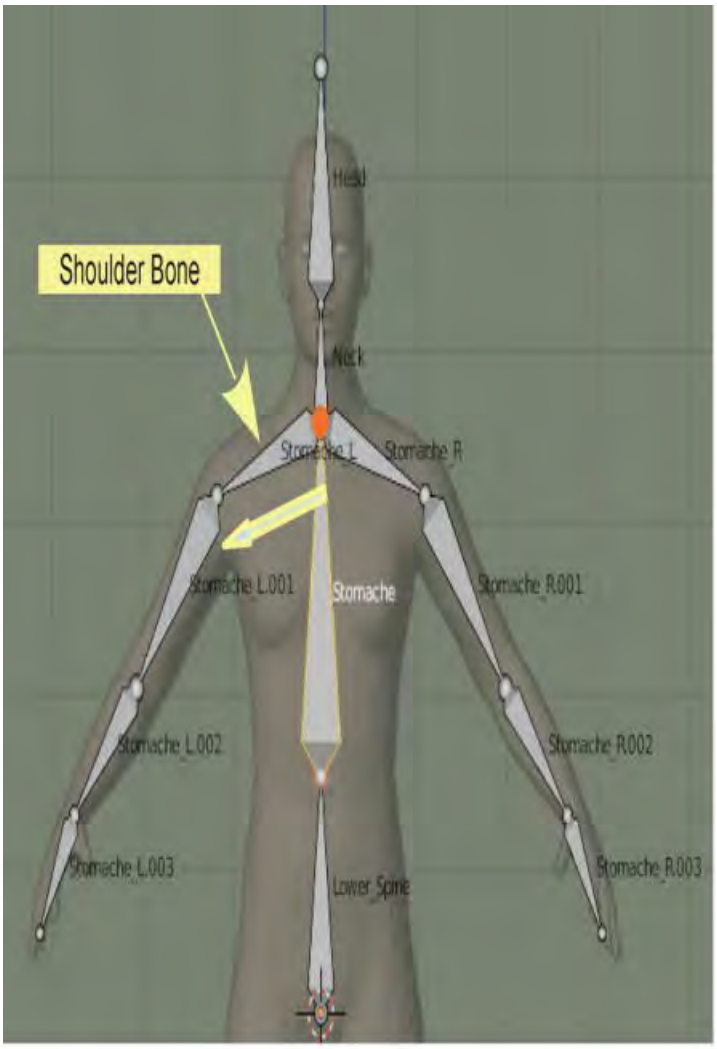
Note: The datablocks in the Outliner Editor containing information about the Make Human Figure.

Figure 22.24

Creating Arms and Hands

To generate the Arm Bonds and Hands start by selecting the Tip of the Stomach Bone (in Edit Mode), G Key (Grab) and position at the top of the chest (Figure 22.25). With the Tip selected E Key (Extrude) the Shoulder Bone.

Note: The Shoulder Bone will be Extruded on one side of the Model only. To Extrude on both sides simultaneously go back a step (Press Ctrl Z). With the Mouse Cursor in the 3D Viewport Editor press the N Key to display the Object Properties Panel and click on the Tool Tab. **Check X- Axis Mirror.**



Object Properties Panel

Click on the Tool Tab. **Check X-Axis Mirror.**

Last Operator Panel

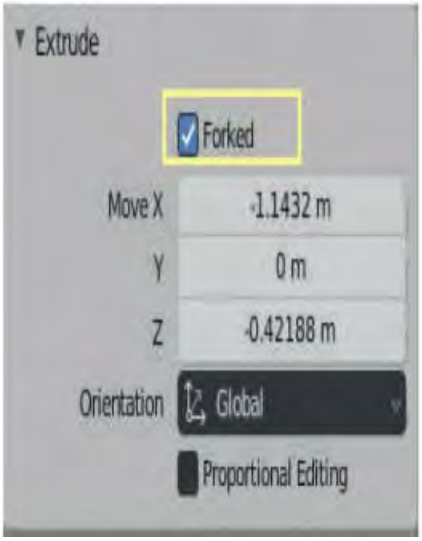


Figure 22.25

The Extrusion will still be on one side only but you will see the Last Operator – Extrude Panel display. Check **Forked** in the Panel. Further Extrusions will take place on both sides.

Continue Extruding the Upper Arm, Lower Arm and Hand.

Note: The Automatic Names. Since the Bones have been Extruded from the Tip of the Stomach Bone they are the children of that bone. Time to rename in the Outliner Editor.

Creating Legs and Feet

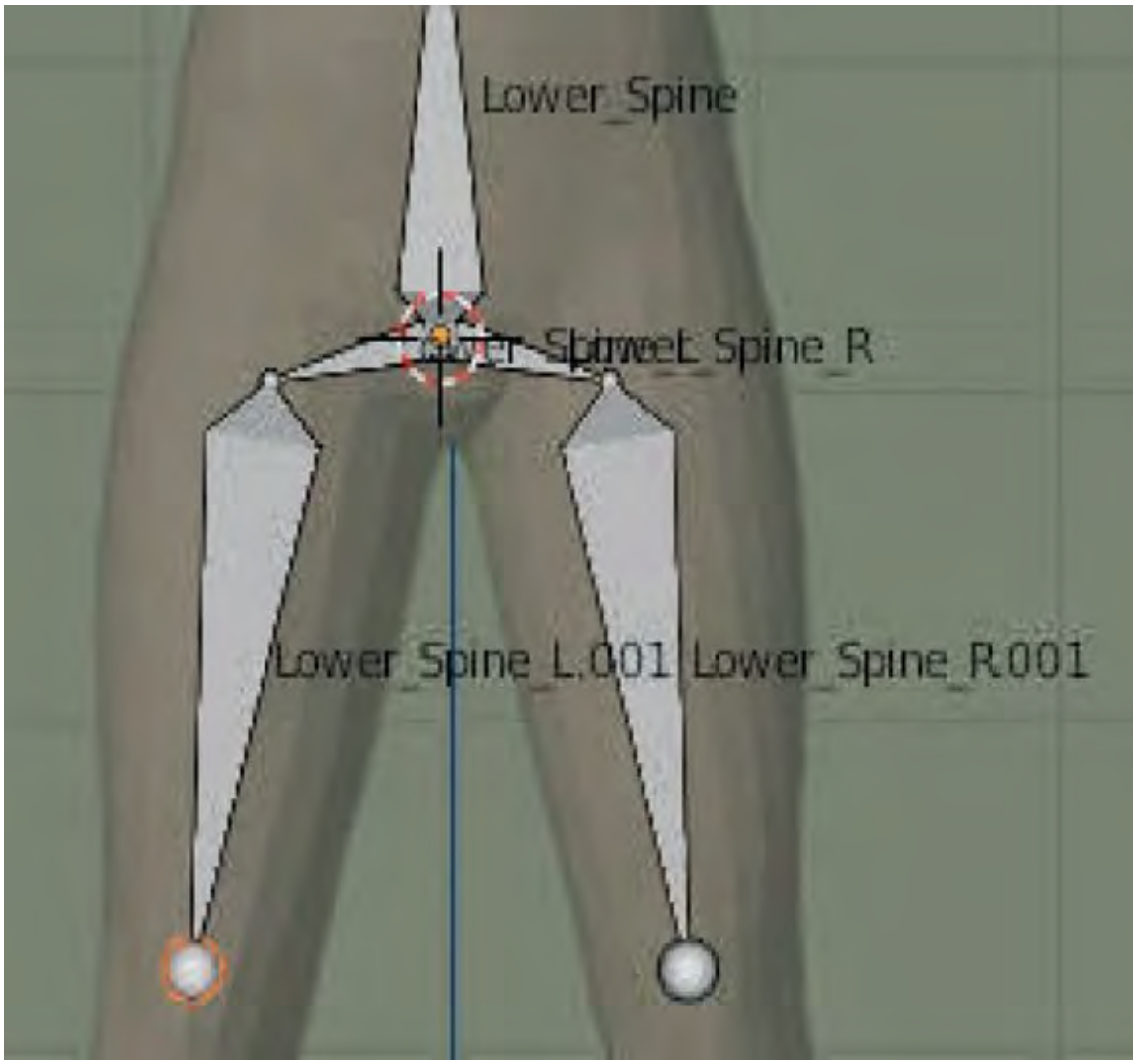


Figure 22.26

Legs and Feet are Extruded from the Base of the Lower Spine Bone using the X-Axis Mirror Function. The first Extrusion should be to generate a Pelvis/Hip Bone.

Note: The automatic naming; Lower_Spine_L, Lower_Spine_R.

Don't forget to incorporate the Left/Right when renaming.

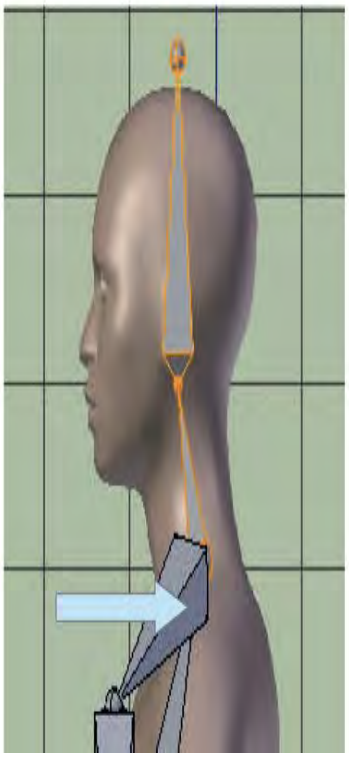
Aligning Bones

In creating the Armature Rig, the construction has been demonstrated in Front Orthographic View. In Right or Left Orthographic Views you will find the

Bones are way out of position relative to the model.

They have to be inside the mesh so when the Bone is Posed the part of the Mesh associated (linked) to the Bone follows the Bone. Select individual Bone Tips/Bases and manoeuvre them into their correct locations (Figure 22.27).

Right Orthographic View – Bones way out of position.



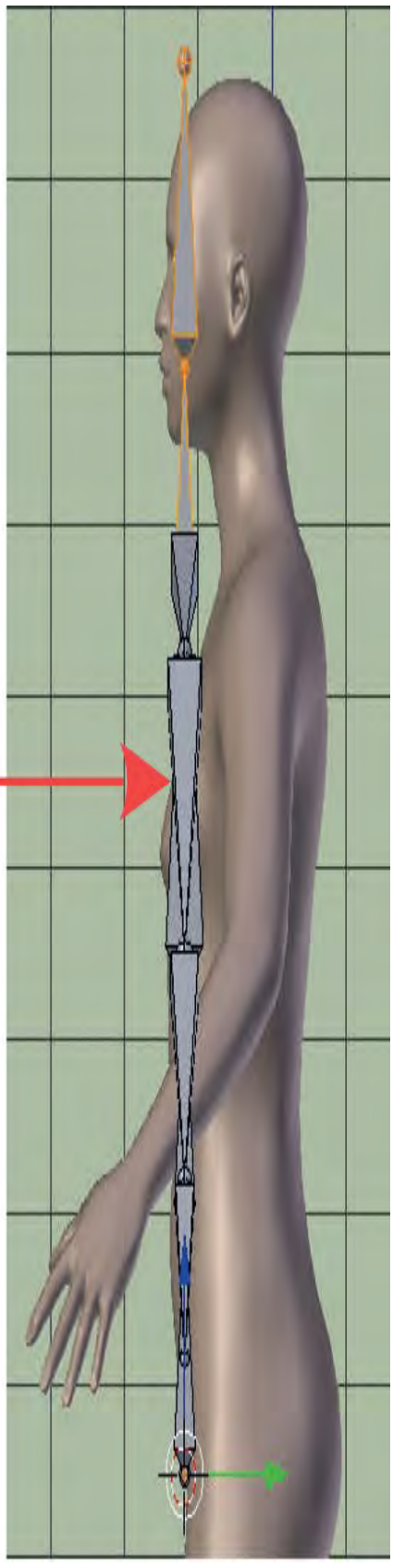


Figure 22.27

Head and Neck Bone correctly positioned

Foot and Hand Bones

The Foot Bones shown in Figure 22.28 begin to demonstrate the detail required in constructing an Armature Rig to Pose and Animate a Model. Hand Bones are even more detailed (Figure 22.9)

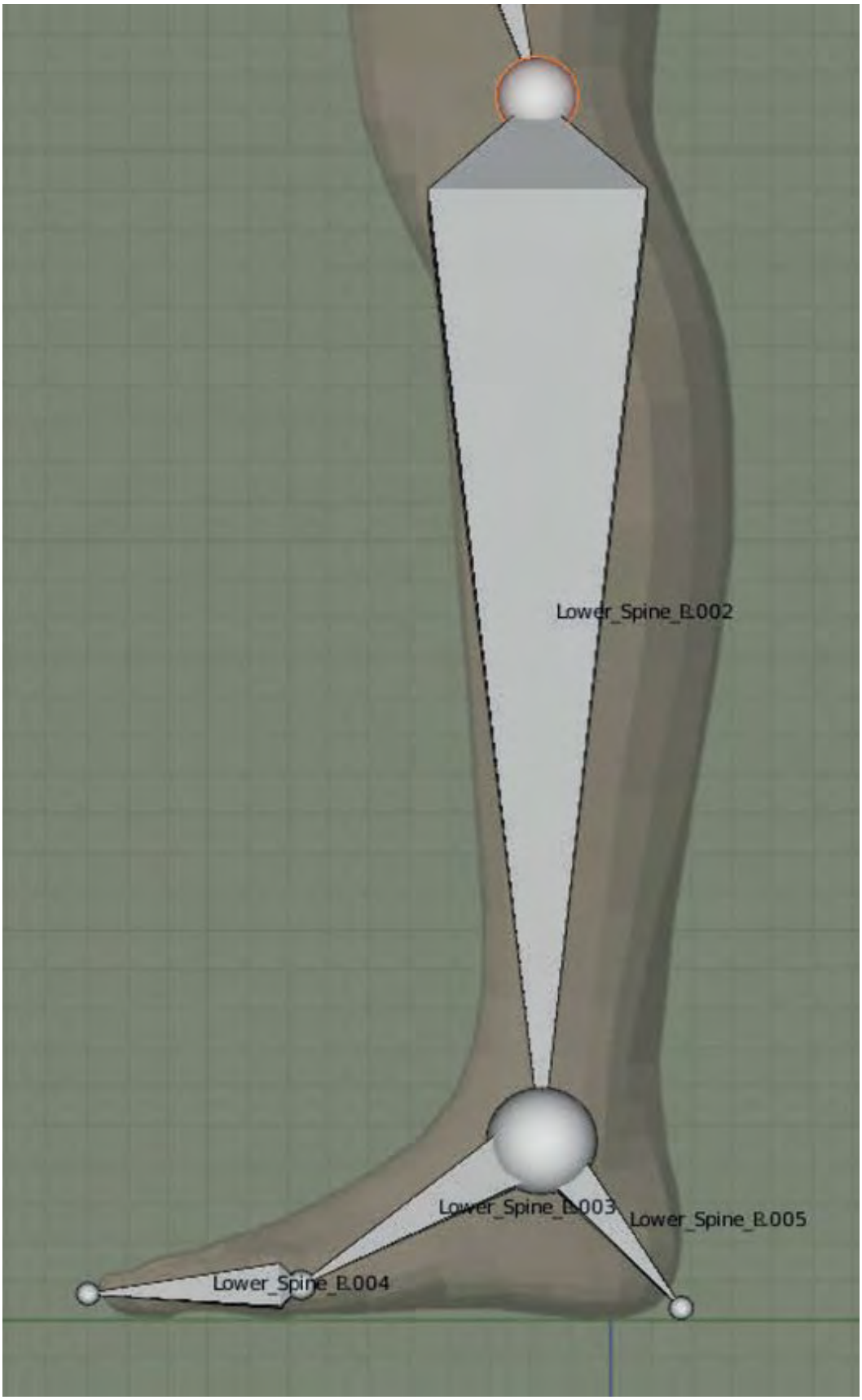
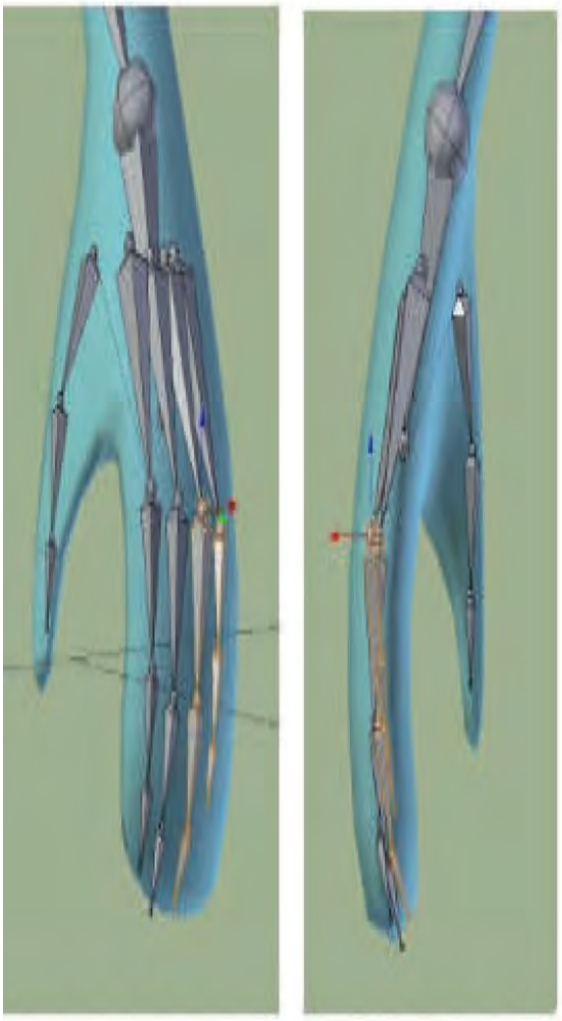

Lower_Spine_B.002
Lower_Spine_B.003
Lower_Spine_B.005
Lower_Spine_B.004

Figure 22.28



Hand Bones inside the Mesh

Bone Axis Displayed

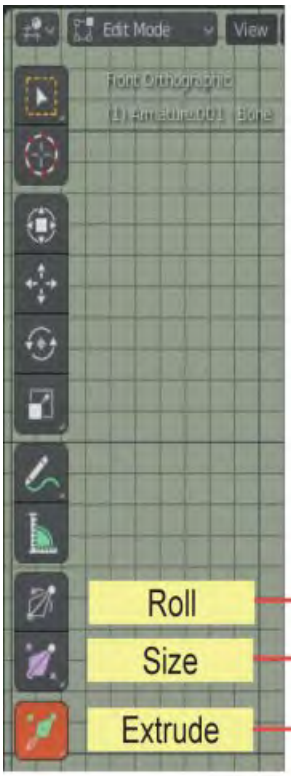
Figure 22.29

Note: In Figure 22.9 the Finger and Thumb Bones appear to be disconnected from the Wrist. This shows that a Bone may be positioned to effect the Mesh without actually being physically connected in the view.

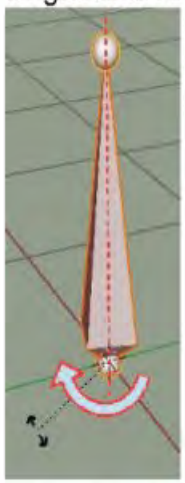
22.9 The Extrude Tool

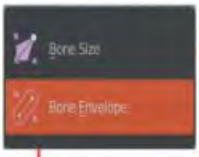
An alternative to pressing the E Key to Extrude a Bone is to click the **Extrude Tool** in the Tool Panel. There are three options; Roll, Size and Extrude. Size and Extrude have sub options.

Note: When an Armature has been added to the Scene and selected in the 3D Viewport Editor, the Edit Mode Tool Panel is an abbreviated version with three Armature Tools.



Roll: Click, hold and drag the Mouse to rotate the Bone about its longitudinal Axis.

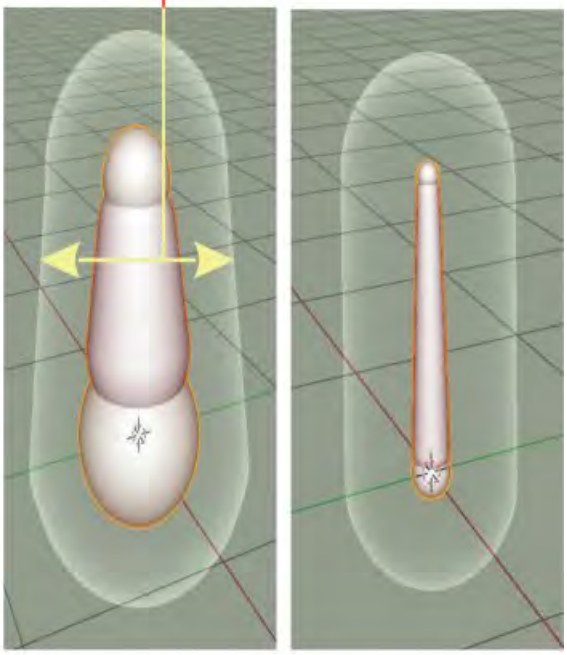




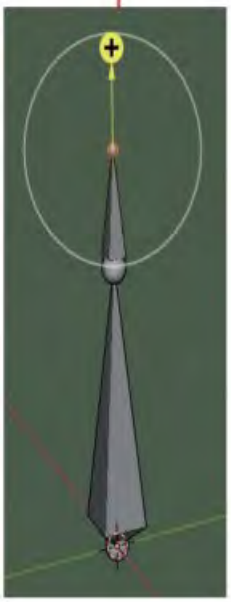
Envelope display Mode

Bone Size difference

Bone Envelope size



Click and hold LMB for options.



Extrude to Cursor

Click LMB in the 3D Viewport Editor. A Bone is extruded from the Tip or Base selected to the position of the Cursor.

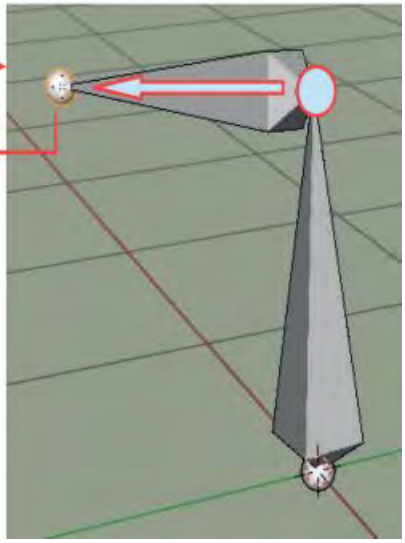


Figure 22.29

Extrude: Click on the cross, hold and drag to extrude in line with the Bone. Click, hold and drag in the circle and Extrude in any direction.

See **Extrude** examples Figure 22.30.

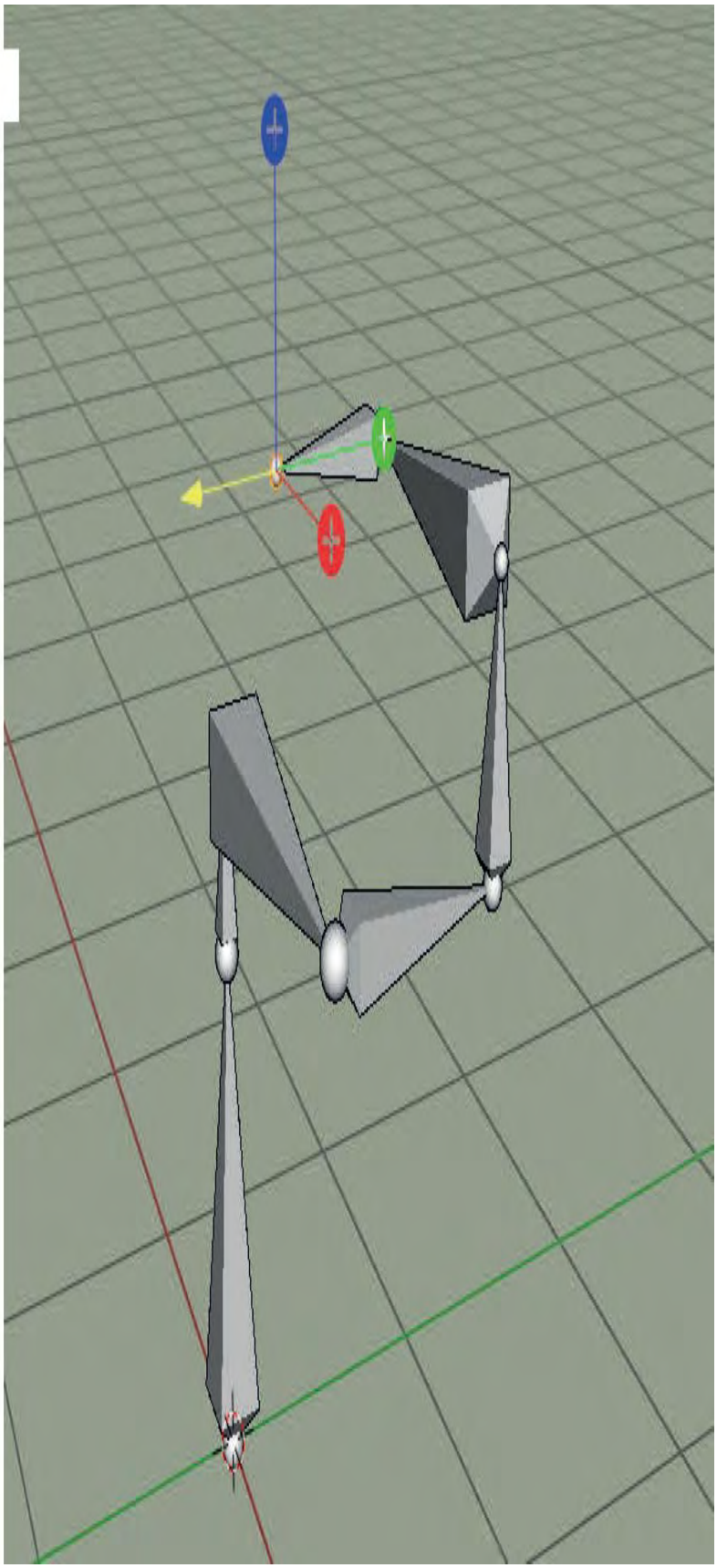
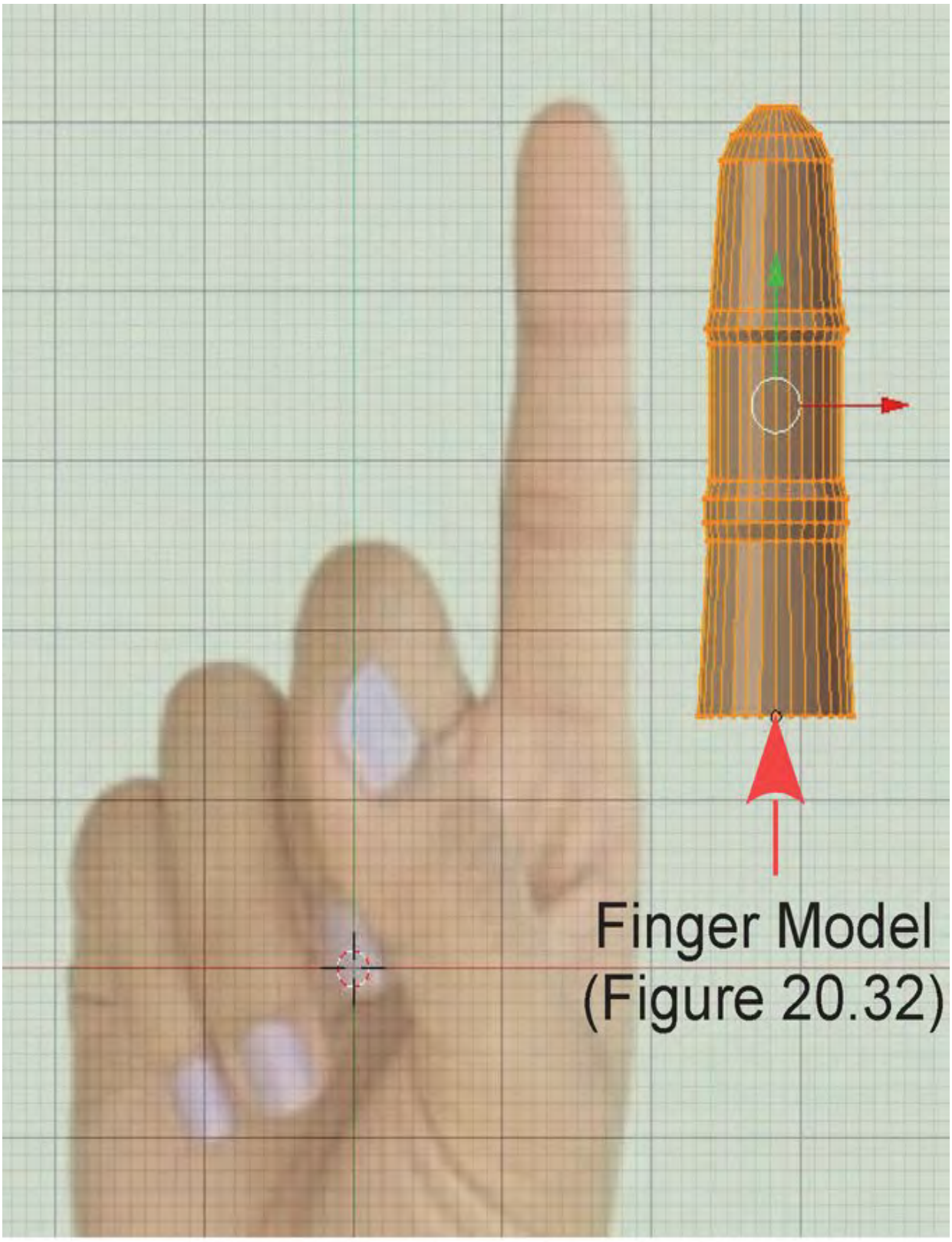


Figure 22.31

22.10 Deforming a Mesh

The basic procedure for deforming a Mesh Object with an Armature is to link or associate vertices on the mesh surface to Bones in the Armature. When the Bones are moved the Vertices in the mesh move.

Make Human models come with a considerable number of Vertices in the surface mesh. That's why they look so good. When deforming a mesh using an Armature you should consider the number of vertices that will be manipulated. A large number of vertices means the computer has to perform a large number of calculations when moving vertices about and that can slow things down considerably. That being the case, in demonstrating mesh deformation, a simple model will be used that has a minimal number of mesh vertices.



Front Orthographic View

Figure 22.32

In Figures 20.32 - 33 a simple finger has been modeled by extruding a Circle. The finger is shown in Edit Mode in Figure 20.32 displaying its Vertices.

Arrange the finger model so that it is pointing up in **Front Orthographic** view with its center of rotation on the center of the Scene.

Construct a three Bone Armature as shown in Figure 22.33.

Place the Armature inside the model.

Parenting

With the Armature in place you **Parent** (associate) parts of the mesh with the Bones by using an **Armature Modifier** or using the **Set Parent To** menu.

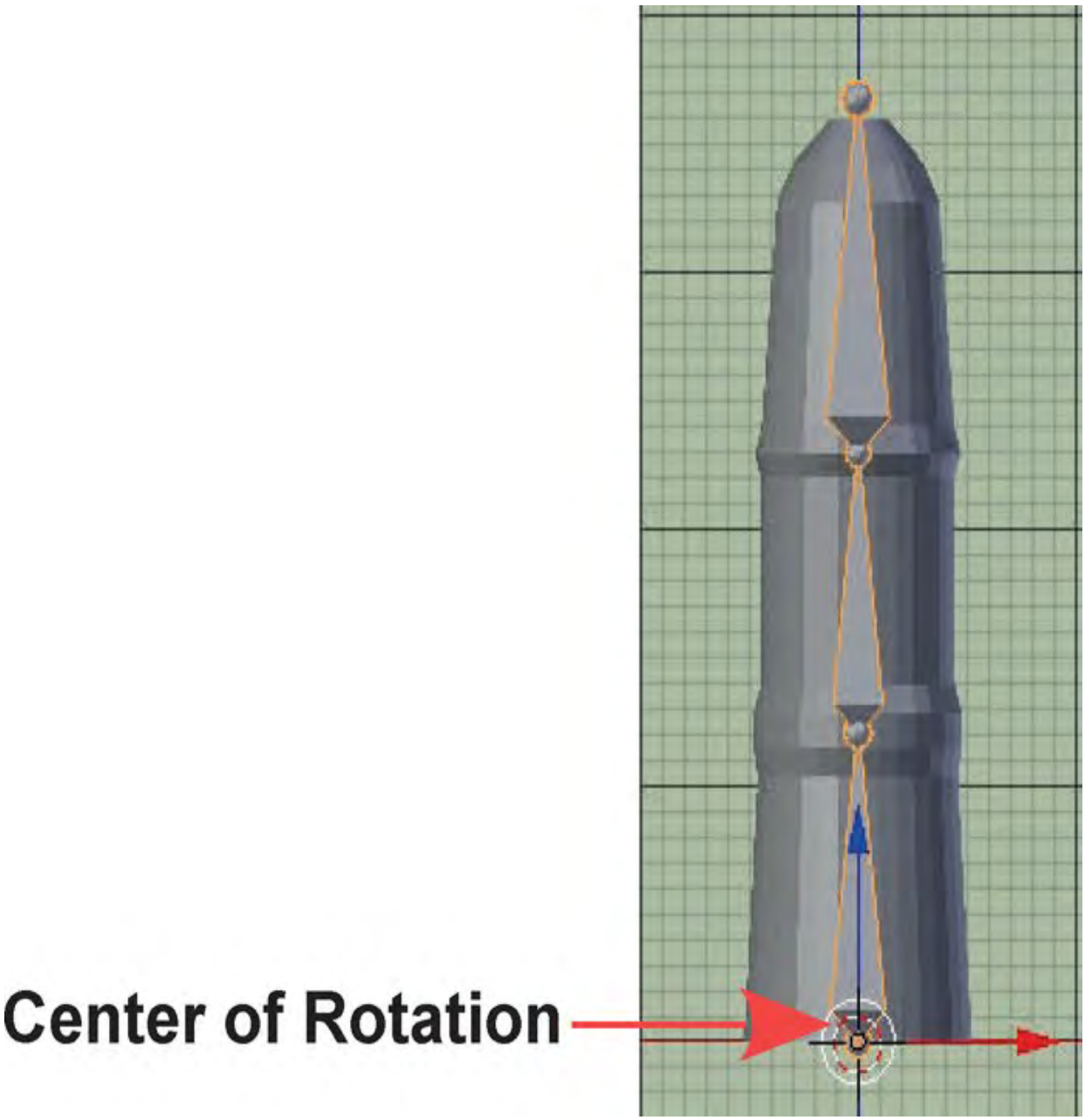


Figure 22.33

22.11 Parenting with The Armature Modifier

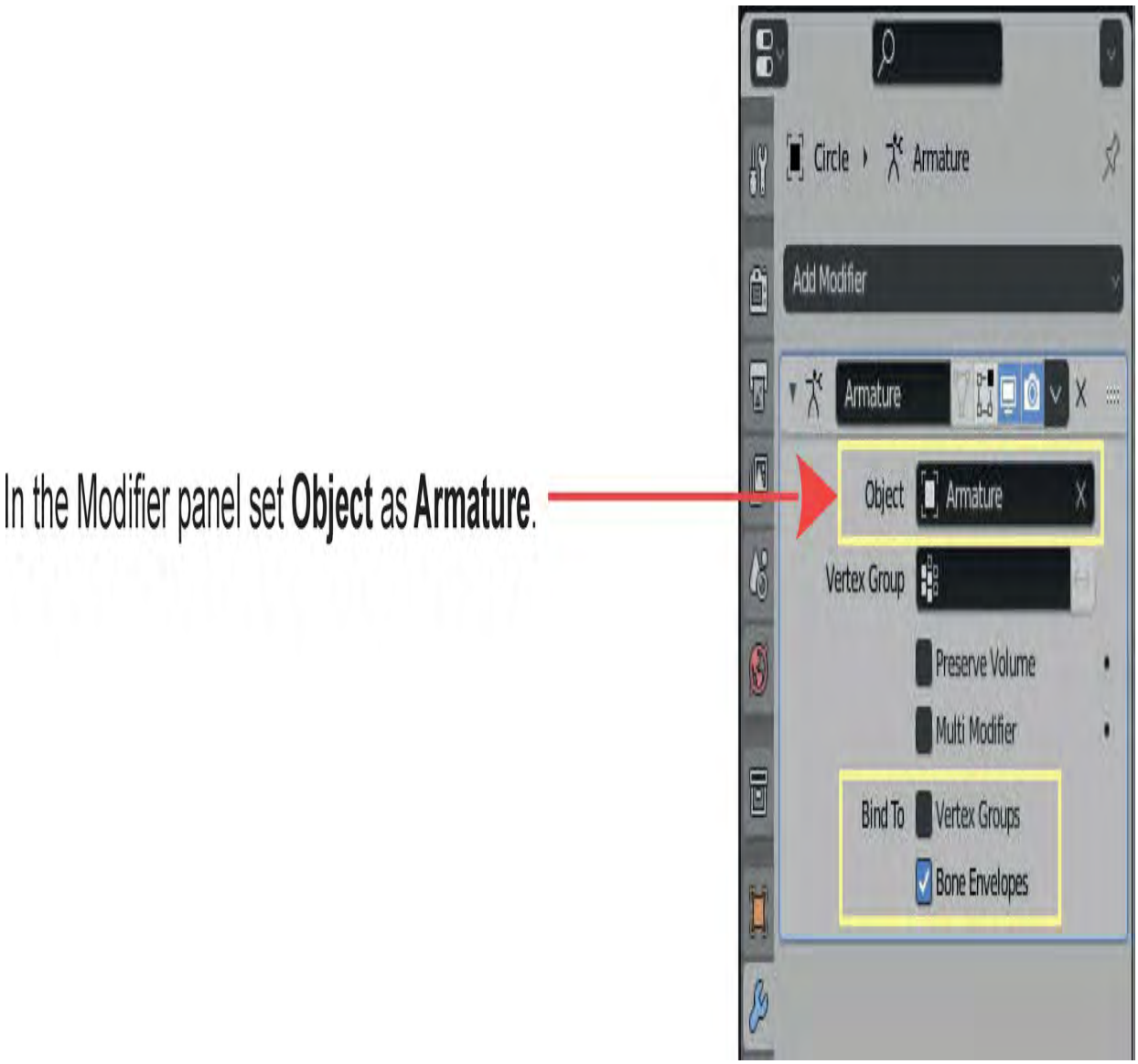


Figure 22.34

In **Object Mode** select the Mesh Model then in the **Properties Editor, Modifier Properties,** add an **Armature Modifier** (Figure 20.34).

In the modifier panel, uncheck **Vertex Groups** and check **Bone Envelope** under the **Bind To** heading. This is telling Blender to associate the Armature Bones with Vertices that are enclosed by the Bone Envelopes (Field of Influence Figure 20.33).

To see what this means have the Armature selected in the 3D Viewport Editor then in the **Properties Editor, Object Data Properties, Viewport Display Tab** change **Octahedral** display to **Envelope** display (Figure 20.35). In the Viewport Display Tab check **In Front.**

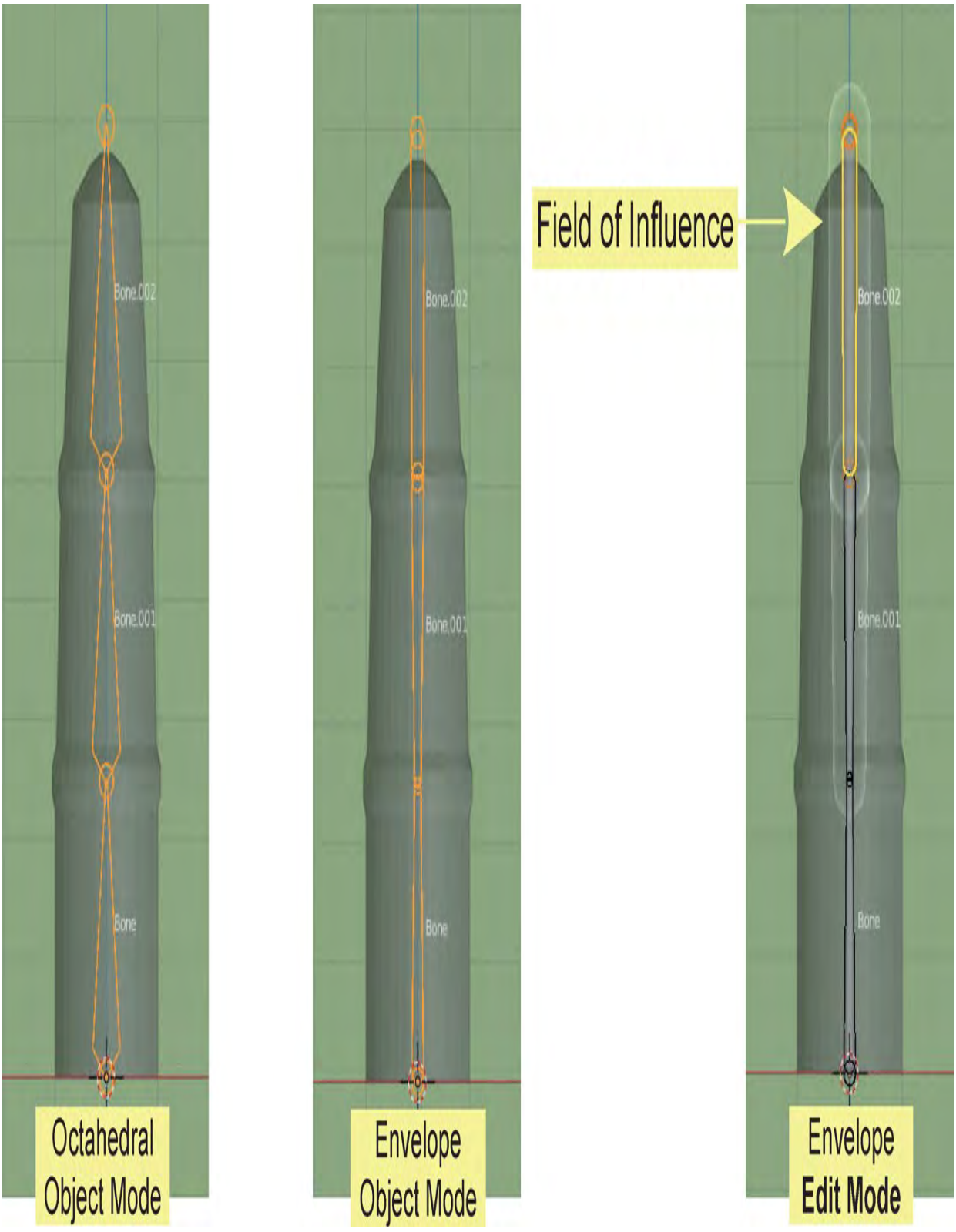
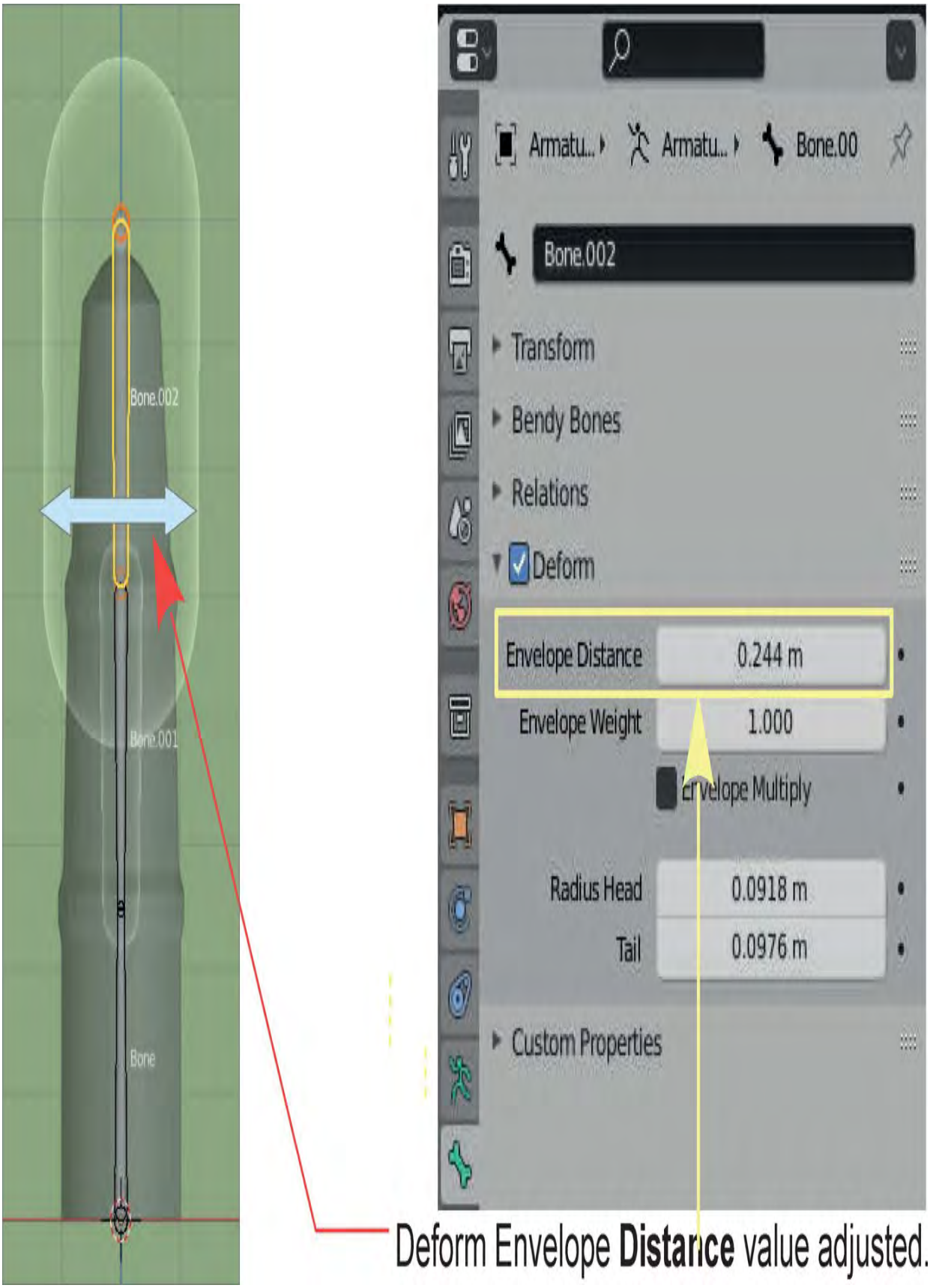

Bone.002
Bone.001
Bone
Octahedral
Object Mode
Bone.002
Bone.001
Bone
Envelope
Object Mode
Field of Influence
Bone.002
Bone.001
Bone
Envelope
Edit Mode

Figure 22.35

Have the **Armature selected** in **Edit Mode.** In the diagram you see the upper finger Bone (Bone.002) selected in Edit Mode and with Envelope Display Mode you see the Field of Influence. This field provides a guide, approximating, which part of the Mesh will be influenced when the Armature is moved. The dimensions of the envelope are adjusted in the Envelope Distance panel, in the Properties Editor, Bone Properties buttons, Deform Tab with the Armature in Edit Mode or Pose Mode (ensure **Deform** is checked in the Tab - Figure 20.36).



Deform Envelope **Distance** value adjusted.

Figure 22.36

With the upper Armature Bone selected in Edit Mode, change to Pose Mode in the 3D Viewport Editor Header. The Bone displays colored blue (Figure 20.37). You may select any Bone and Rotate (R key drag mouse) to see the mesh deform (Figure 20.38).

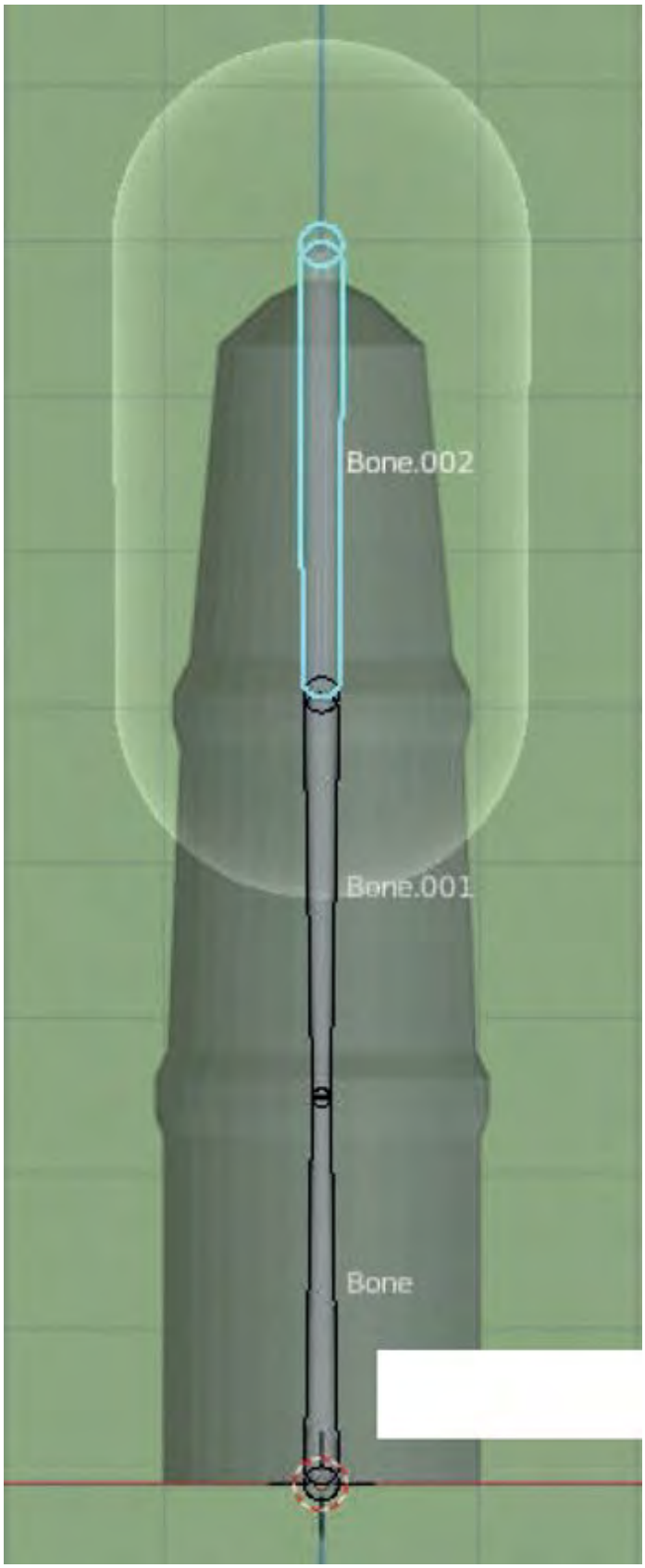

Bone.002
Bone.001
Bone

Figure 22.37

Bone.002
Bone.001
Bone

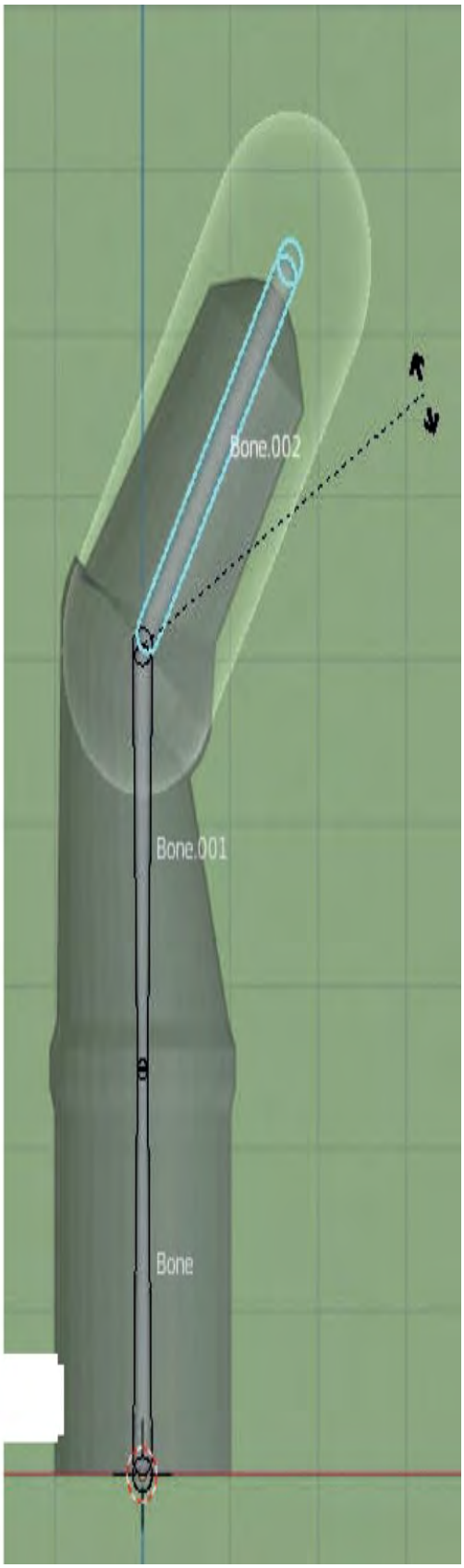

Bone.002
Bone.001
Bone

Figure 22.38

Note: The Bone Envelope Distance has an effect on how the Mesh deforms. This has to be adjusted for the each Bone section. Experiment with the Envelope Distance setting and subdivide the Mesh to achieve the desired deformation.

By selecting the Mesh Model in Edit Mode, in the Properties Editor, Object Data Properties you will see that a Vertex Group has been created for each Bone in the Armature. By setting the **Bind To:** value in the Armature Modifier to **Bone Envelope** (Figure 20.33) the Vertices in the Vertex Groups are controlled by the Bone Envelopes.

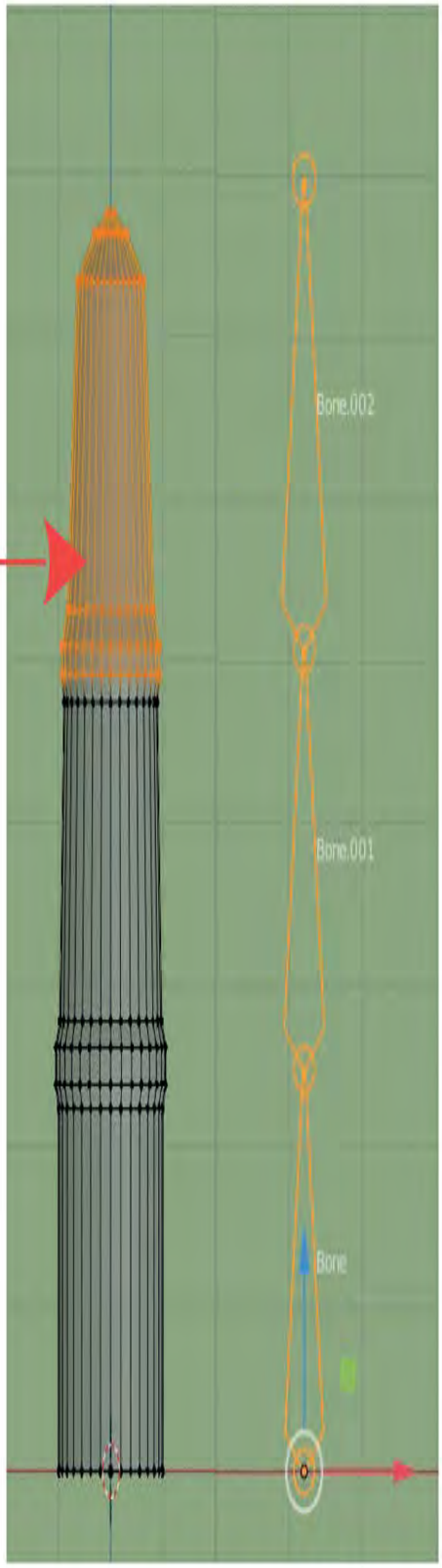
22.12 Assigning Vertices – Vertex Groups

In the previous examples (20.11) **Mesh Vertices** were selected from within the **Field of Influence** of the Armature by the Armature Modifier. An alternative to this, is to manually nominate which vertices will be affected by the Armature.

Use the same Finger – Armature arrangement previously described. Select the **Armature** then in the **Properties Editor, Object Data Properties, Viewport Display Tab,** tick **Names** to show the Bones named **Bone** and **Bone.001** and **Bone.002.** Deselect the Armature.

Select the **Finger** (Mesh – Object Mode), then in **Edit Mode** deselect the Vertices. In the **Properties Editor, Object Data Properties, Vertex Groups Tab,** click the **Plus** sign to add a **Vertex Group;** a Vertex Group is added and named **Group.** By renaming **Group** to **Bone.002** (Figure 22.39), the **Vertex Group** will automatically be controlled by the Bone named **Bone.002.** Groups and Bones may be renamed to whatever you want, but for a Group to be controlled by a Bone, the names must be identical.

A second condition for control is; You must apply an Armature Modifier to the Finger with the Object set as Armature and **Bind To: Vertex Groups** checked.



Vertices assigned to Vertex Group named Bone.002

Armature Bone.002 controls Mesh deformation

Armature shown moved to one side

Object Data Properties

Figure 22.39

In the 3D Viewport Editor, select the Vertices in the upper part of the finger (press the B key – drag a rectangle). Make sure you have the **Toggle X-Ray button turned on** in the 3D Editor Header or will be selecting the front Vertices of the finger.

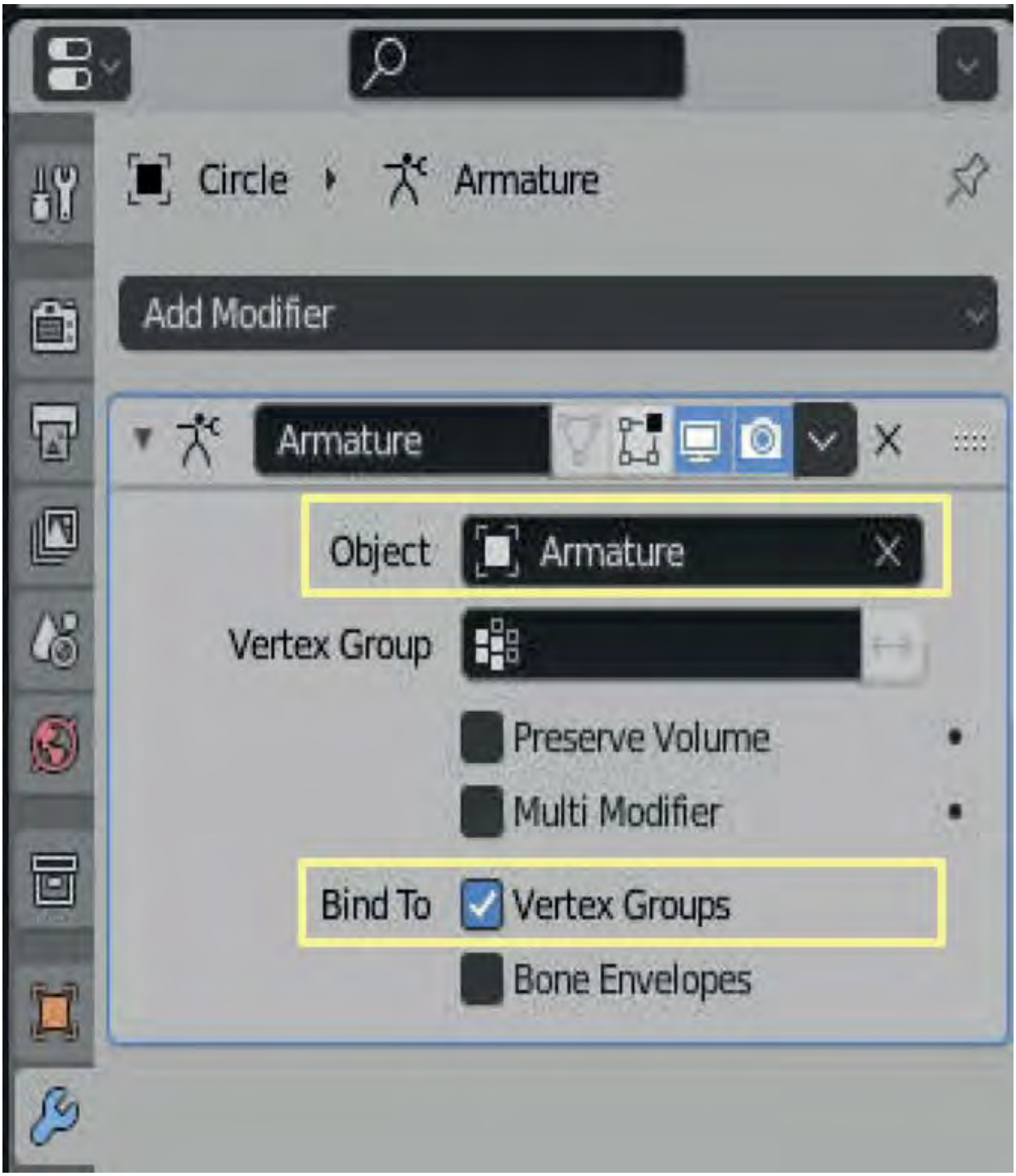


Figure 22.40

In the **Vertex Groups** tab, click **Assign** to assign the selected Vertices to the Group. Check out the assignment by alternately clicking on **Deselect** and **Select** (Figure 20.39).

Tab into **Object mode** and deselect the finger. Select the **Armature** and change to **Pose Mode.** Select **Bone.002** and press the **R Key** to rotate. **Nothing happens if you haven't applied an Armature Modifier to the finger** (Figure 22.40).

Go back and select the finger and in the **Properties Editor, Modifier buttons,** click **Add Modifier** and select **Armature.** In the **Armature Object** panel, click and select **Armature.**

Deselect the finger and select the **Armature** in **Pose Mode.** Select **Bone.002** and rotate it–the upper part of the finger will now deform as the Bone is rotated (Figure 22.41 -42).

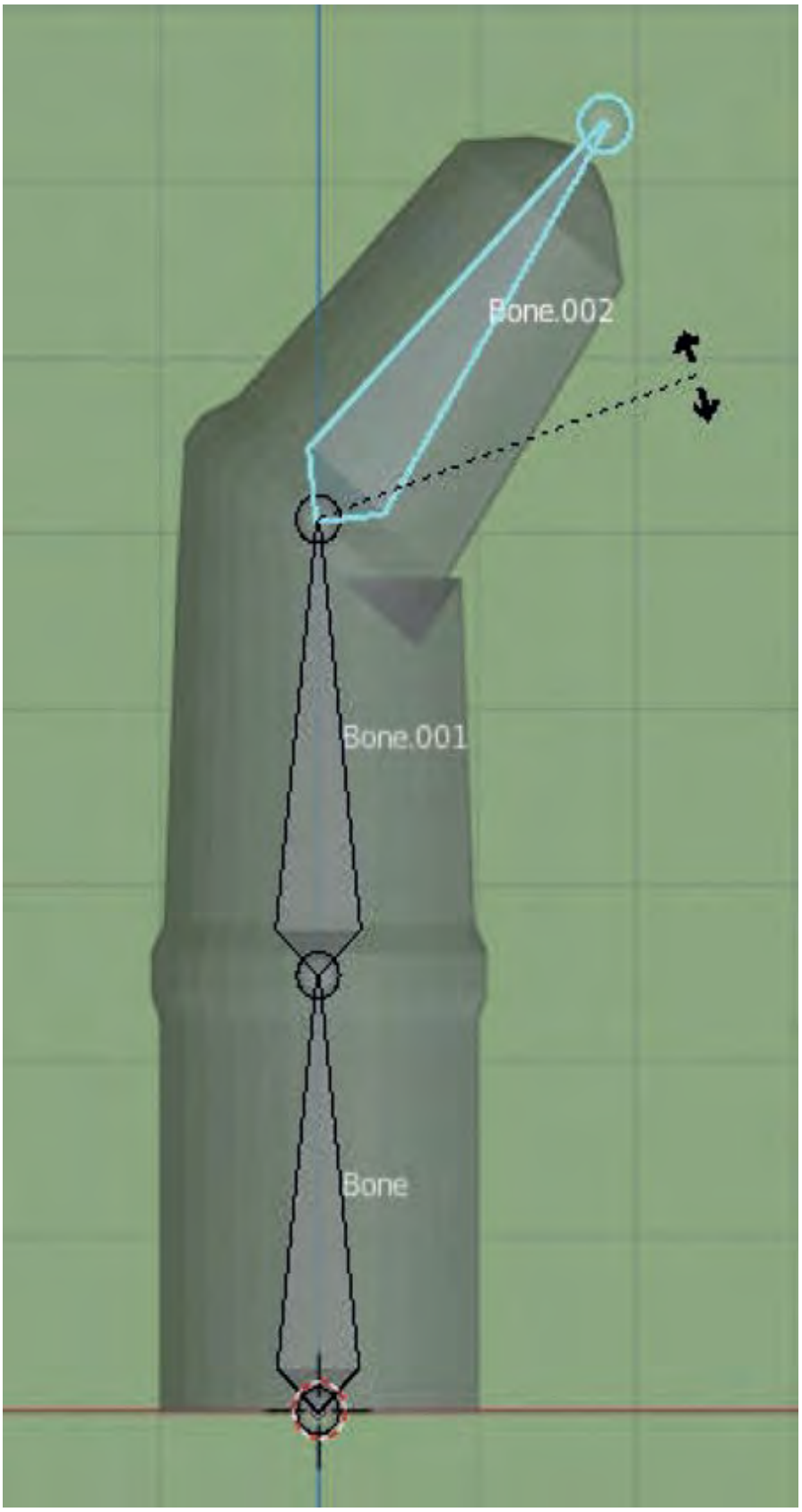

Bone.002
Bone.001
Bone

Figure 22.41

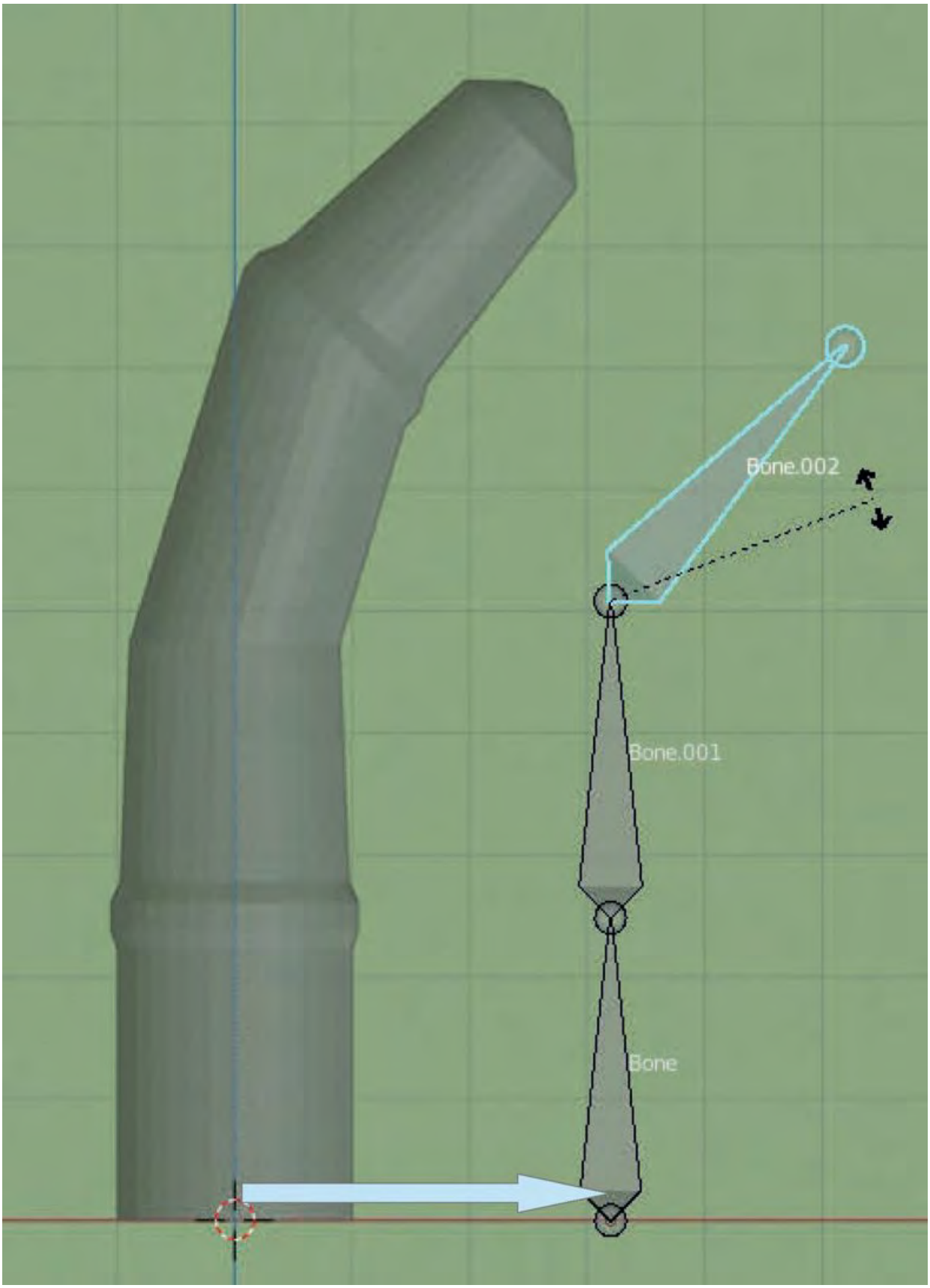

Bone.002
Bone.001
Bone

Figure 22.42

The armature may be located well away from the finger and still deform the Mesh. The Field of Influence of the Armature described in the previous exercise is not enforced, but with the Armature displaced away from the Mesh, the Mesh deformation is exaggerated (Figure 22.42).

The foregoing has outlined the manual procedure for creating a Vertex Group and employing an Armature Modifier.

Note: In the Armature Modifier, **Object: is set as Armature** and **Bind To; is set as Vertex Groups.** In the Object Data Properties for the Mesh Model the single Vertex Group is named the same as the controlling Bone (Bone.002).

As you can image, in a complicated Mesh/Armature arrangement manually creating and assigning vertices to group fore each Bone would be a tedious task. That being the case, Blender incorporates semi automatic process. This is demonstrated using the same Mesh Finger with the three Bone Armature.

Note: Be aware that when using these processes adjustments may be required to refine the control.

The automated processes are selected from the Set Parent To menu.

22.13 Assigning Vertices – bet Parent To Menu

In the Mesh Finger – Armature arrangement (Figure 22.43), in Object Mode, select the Mesh then Shift select the Armature. Press Ctrl + P key and in the **Set Parent To** menu that displays chose (click) one of the **Armature Deform** options.

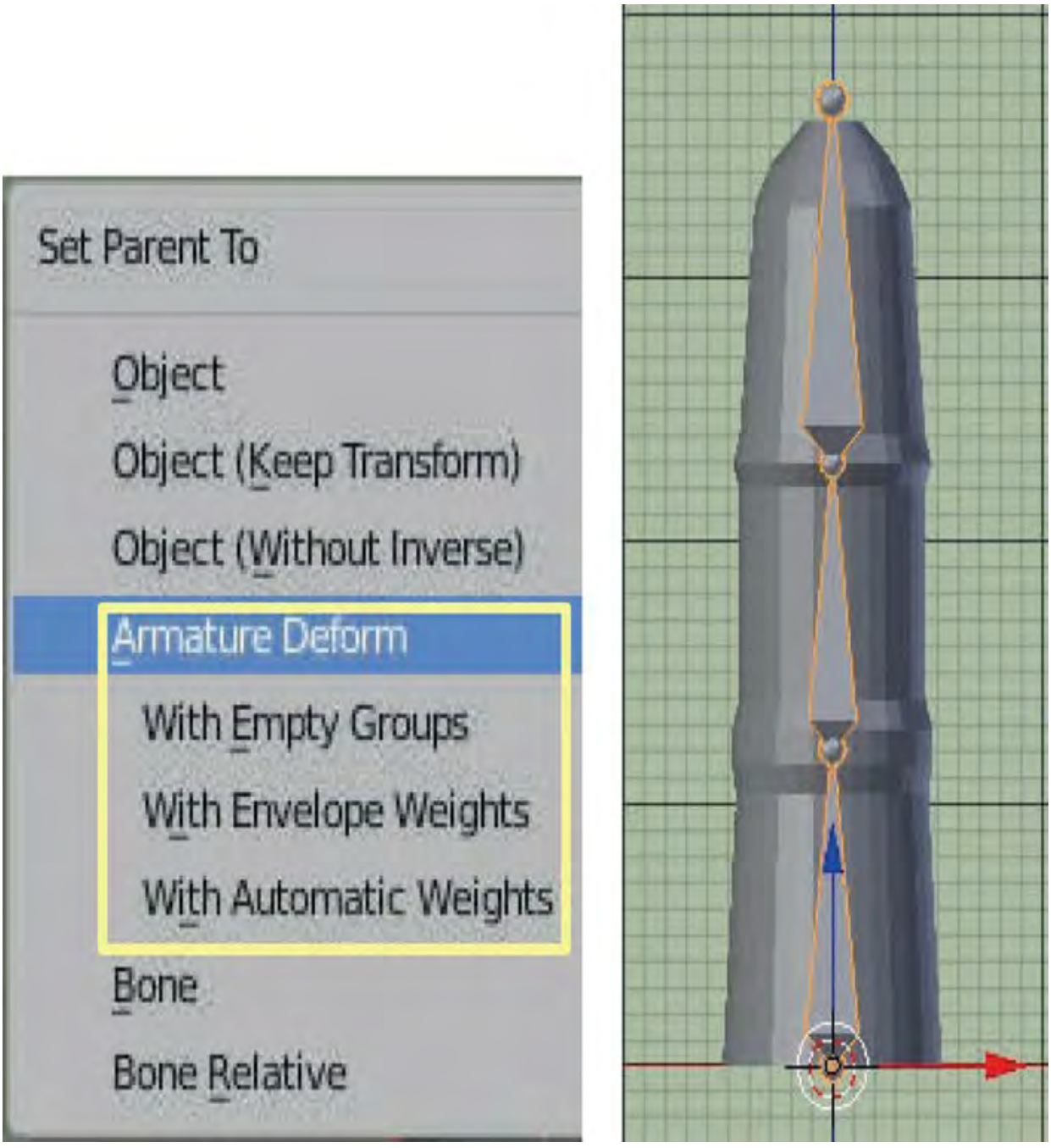


Figure 22.43

With Empty Groups creates a child parent relationship between the Mesh Object and the Armature such that when a Bone in the Armature is moved in Pose Mode part of the Mesh follows. This is accomplished by automatically creating an **Armature Modifier** for the Mesh with **Object: Armature** and

Bind To: Vertex Groups (see the Modifier Properties). At the same time Vertex Groups are created and named in accordance with the Bones in the Armature (see the Object Data Properties). Both sets of Properties are selected in the Properties Editor with the Mesh Object selected in the 3D Viewport Editor in Object Mode.

Note: Although Vertex Groups are created and named in accordance with the Bones in the Armature, Vertices **have not** been assigned to the Vertex Groups. Hence **With Empty Groups.**

With the Mesh Object in Edit Mode, select Vertices and assign to each Vertex Group.

Moving a Bone in the Armature, in Pose Mode, will see the assigned Vertices follow the movement.

With Envelope Weight creates the same Armature Modifier and Vertex Groups but this time you change **Bind To** from **Vertex Groups** to **Bone Envelopes** in the Armature Modifier and the Envelope will control which Vertices move when the Bone is moved in Pose Mode. You may alter the Bone Envelope size to adjust the control.

With Automatic Weights, again, creates an Armature Modifier and Vertex Groups but this time Vertices are automatically assigned to the Vertex Groups. The automatic assignment may or may not create the desired effect, therefore, you manually correct the assignment of vertices to the Vertex Groups.

22.14 Assigning Vertices – Weight Paint

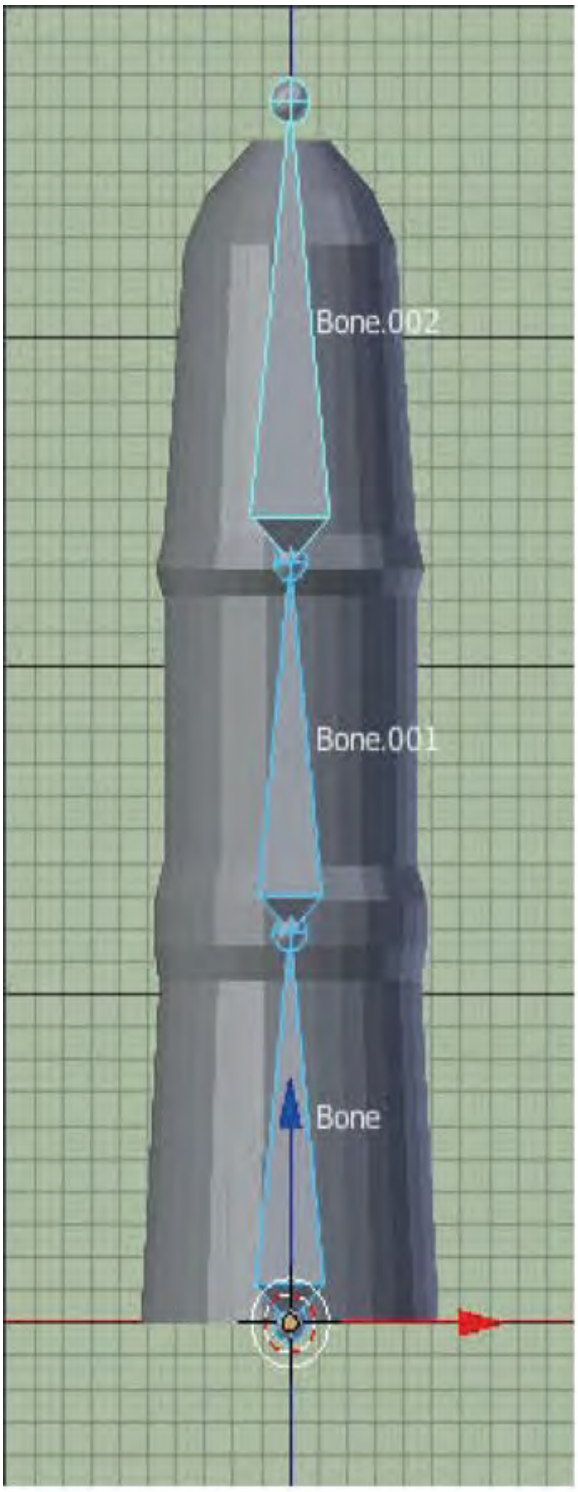
Blender has a painting method that selects and assigns Vertices to a group, automatically linking to an armature Bone. The Paint method allows a graduated weight to be given to vertices that dictate how much influence the armature Bone will have over the deformation of the Mesh.

Set up a new Scene as you did for the previous examples. Select the **Finger** in **Object Mode** and add an **Armature Modifier** in the Properties Editor.

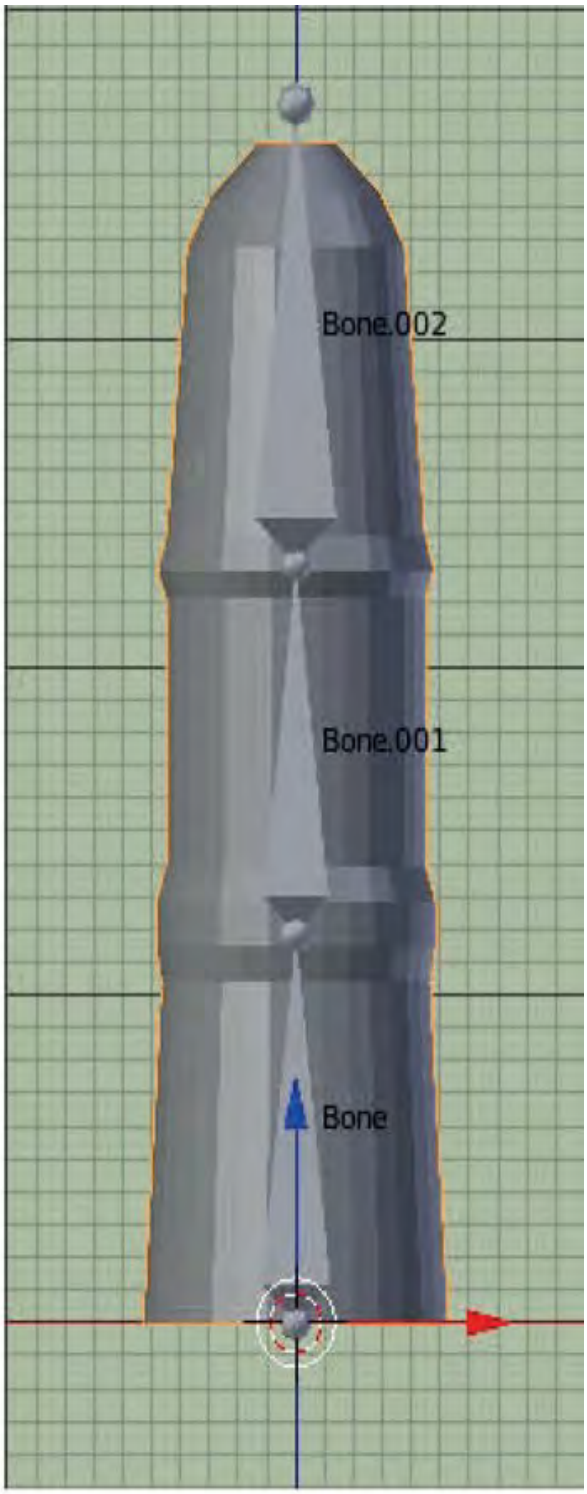
Don’t forget to enter **Armature** in the **Object panel.** Select the Armature and enter **Pose Mode.** In the **Properties Editor, Object Data Properties, Viewport Display Tab,** check **Names** to display the Bone names in the 3D Viewport Editor; the names should be **Bone, Bone.001** and **Bone.002** as before.

When ticking **Names** make sure you are in the **Object Data Properties** not the **Object Properties.** If you are in the Object Properties only the name **Armature** will display.

With the Armature as the selected Object in the 3D Viewport, in Pose Mode, select **Bone.002** (the upper Bone) then change to Object Mode and left click the Finger to select it. With the finger selected, go to the 3D Viewport Header and change from Object Mode to **Weight Paint Mode.** The Finger displays dark blue, indicating that no Vertices are selected (Figure 22.44).



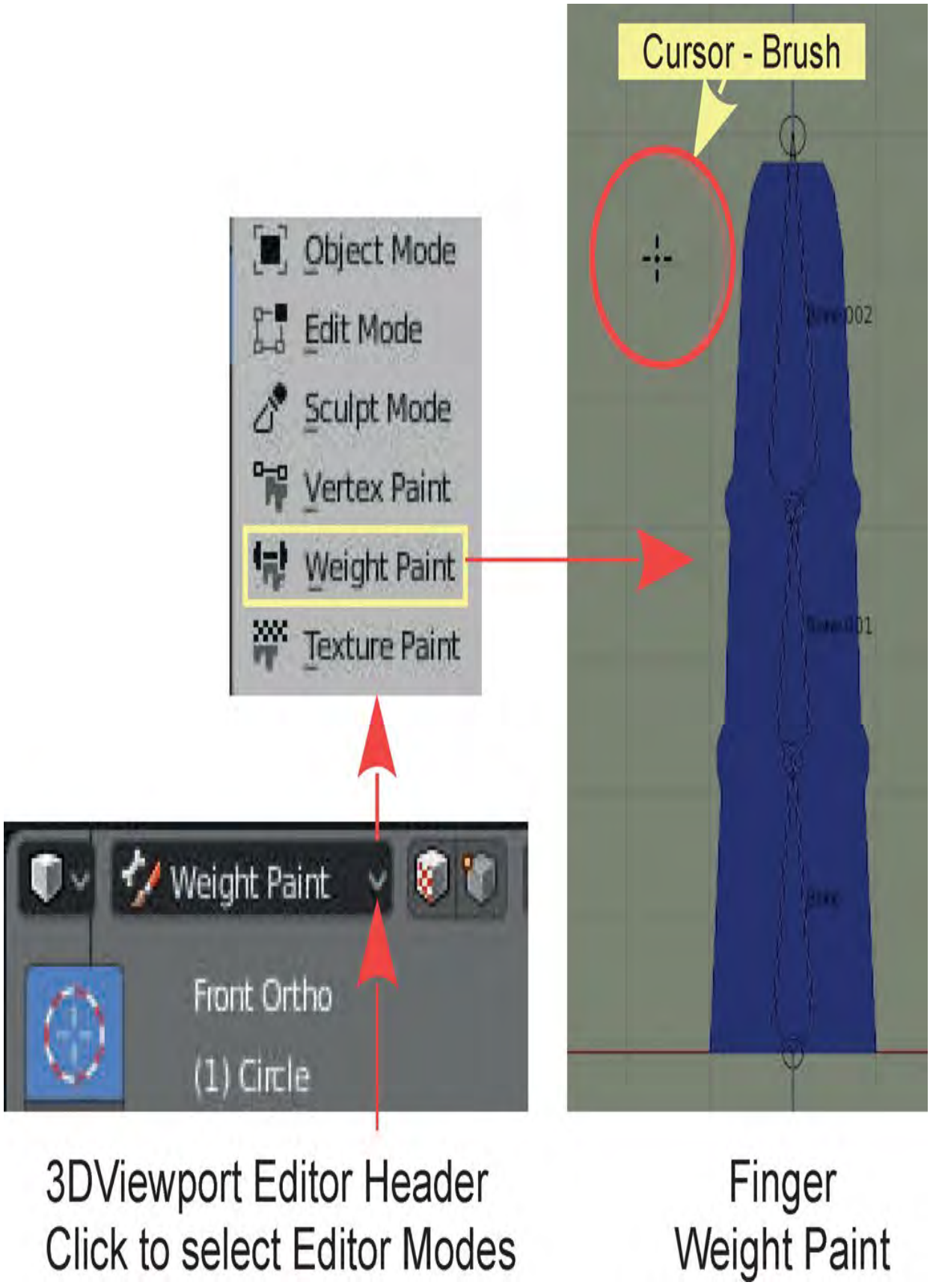
Armature
Pose Mode



Finger
Object Mode

Figure 22.44

The Finger displays dark blue, which indicates that no Vertices are selected.



3DViewport Editor Header
Click to select Editor Modes

Finger
Weight Paint

Figure 22.45

In the **Tool Panel** at the left-hand side of the 3D Viewport have the **Draw Tool** selected. In the Header click **View** and check **Tool Settings.** Make sure the **Weight** and **Strength** sliders are set to 1.000 in the Header (Figure 22.46). You are about to paint over the finger mesh to select Vertices, and by setting the strength to a high value you are telling Blender that the selected Vertices are to be rigorously controlled by Bone.002. In Weight Paint Mode, the cursor in the 3D Viewport Editor is a circle (Figure 22.46). The Radius control for the circle is in the Header. You want the upper part of the finger to be transformed by Bone.002, therefore, click, hold, and drag the Cursor circle over the upper part of the Finger. As with selecting vertices make sure **Toggle X-Ray** is turned on in the Header.

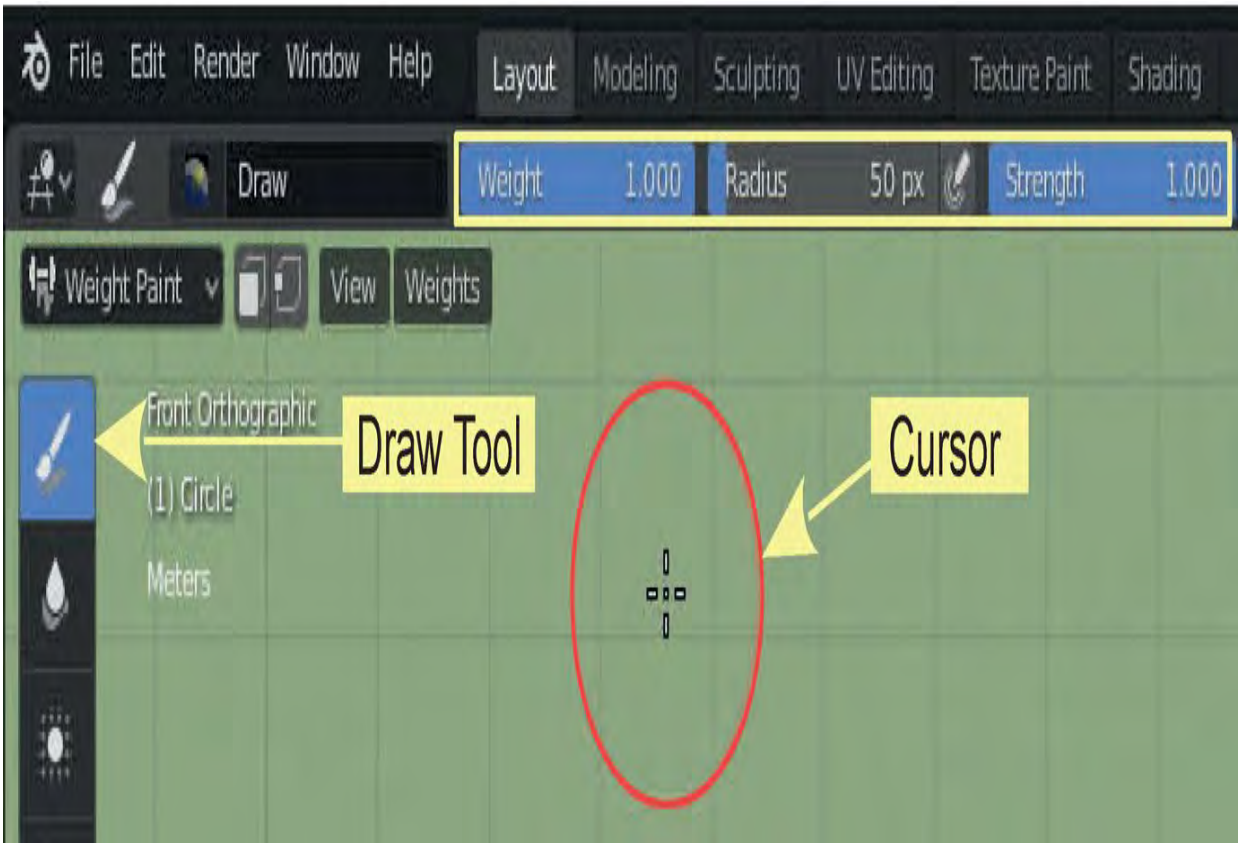


Figure 22.46

The part of the Finger painted turns red, which indicates a rigorous control (Figure 22.47). Altering the **Strength** value changes the control strength and will display a different color.

Turn the mesh around and paint the Vertices on the back side of the finger (pan the 3D Viewport around).

Vertices Assigned (Red)

Vertices Not Assigned
(Blue)

Figure 22.47

Having painted the finger, note that in the **Properties Editor, Data buttons, Vertex Groups Tab** a **Vertex Group** has been created with the default name: **Group.**

Important: Rename the Vertex Group: **Bone.002.**

Selecting **Bone.002** in **Pose Mode** and rotating it will move the upper part of the finger (Figure 22.48). Repeat the process for the remaining Bones.

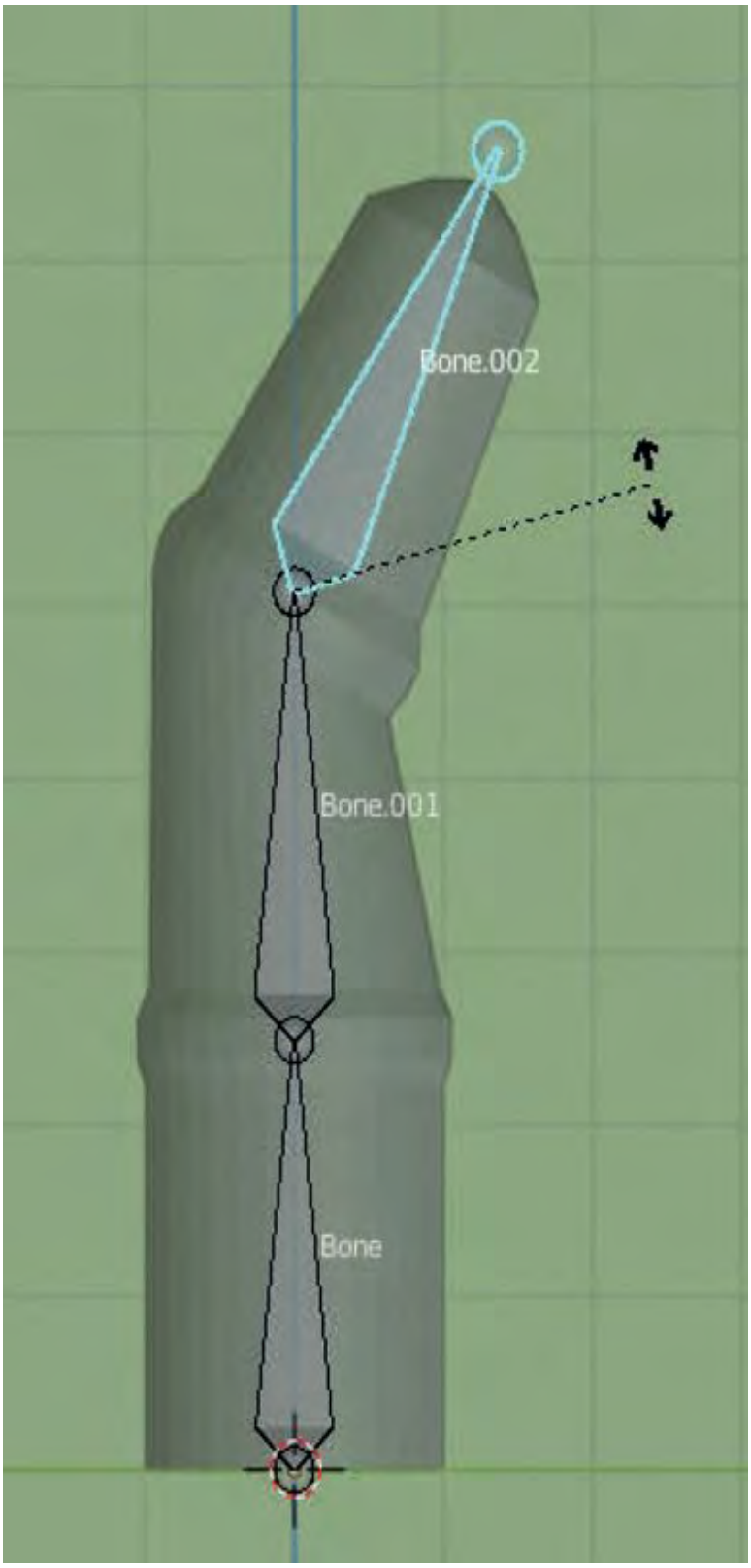


Figure 22.48

With vertices assigned to the different Bones in the Armature, when individual Bones are manipulated the Vertices follow reshaping the mesh.

There are occasions when you will want several Bones to follow a single Bone. For example; Translating Bone.002 and having Bone.001 and Bone follow the movement. In this instance you would employ the **Inverse Kinematic Constraint.**

22.15 Inverse Kinematics Constraint

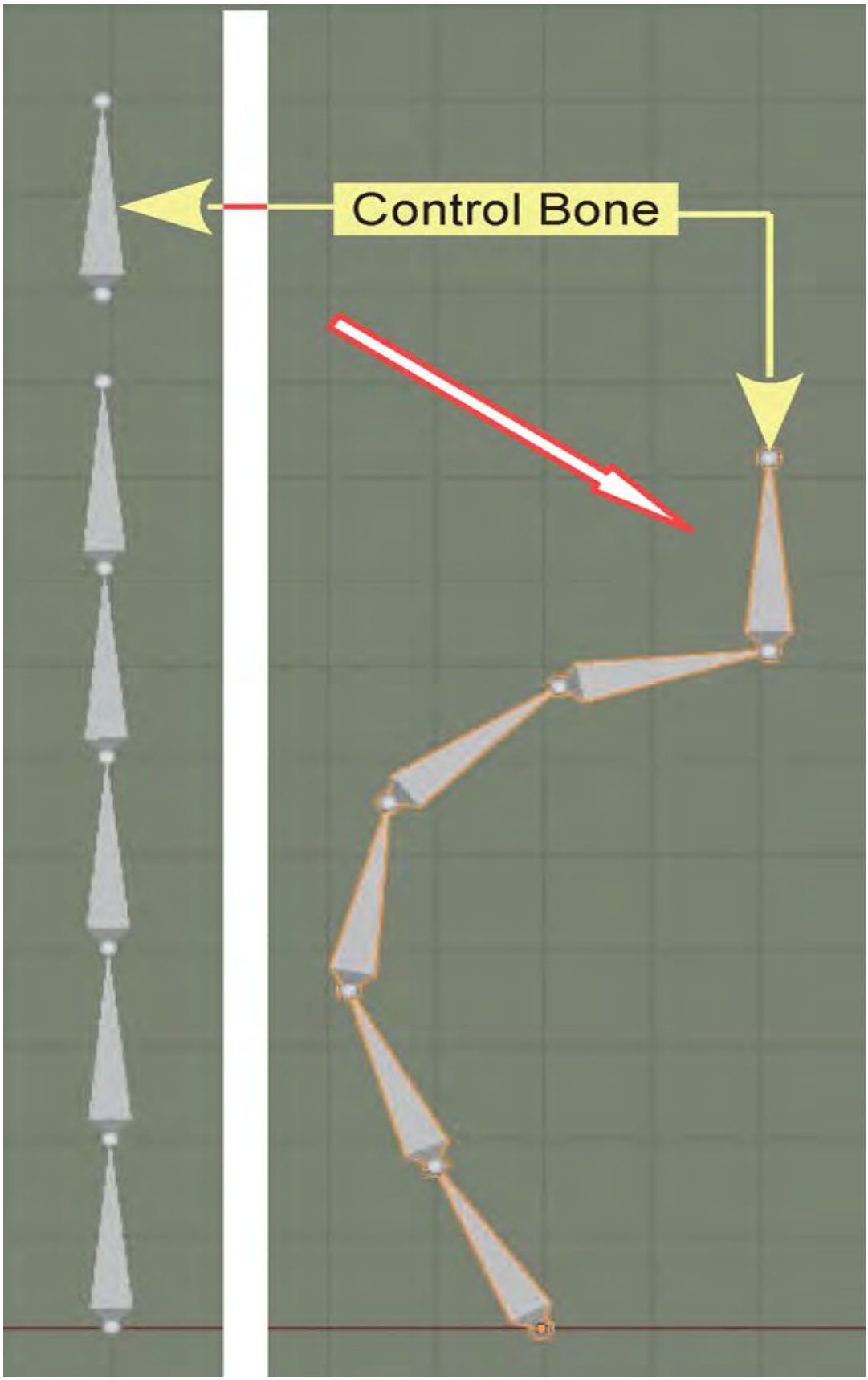

Control Bone

Figure 22.49

The **Inverse Kinematics (IK) Constraint** is used for controlling a chain of Bones (Figure 22.49).

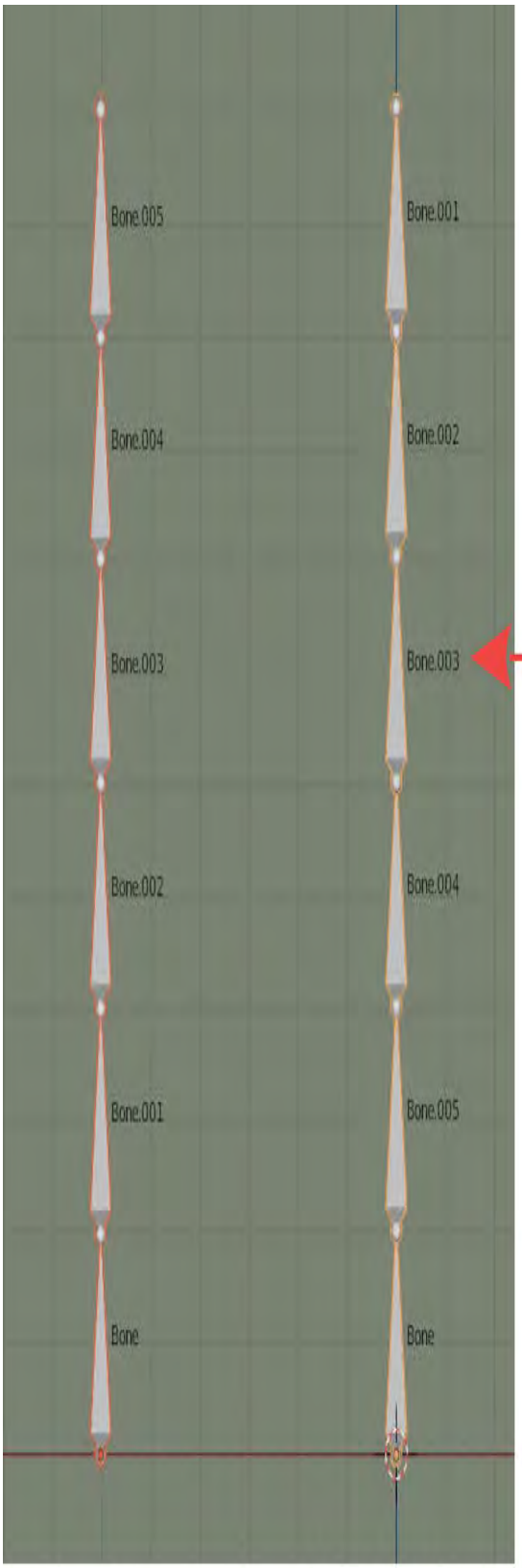
With IK, dragging a **Control Bone** will result in the chain following the Control.

To Pose the Chain, without **Inverse Kinematics** you have to Pose individual Bones one by one; this is a tedious process but gives you full control.

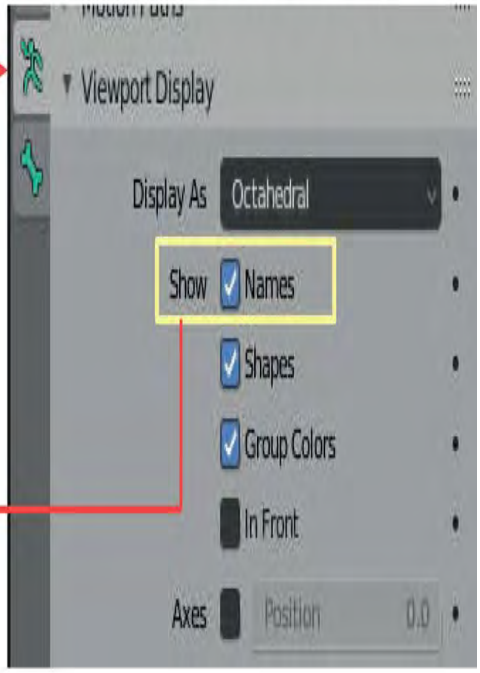
Do not confuse **IK (Inverse Kinematics)** with the **Spline IK.**

An example of **Inverse Kinematics** would be to create a chain of Bones (Figure 22.49). **Note: Figure 22.50 shows two separate chains.** One chain has been created by extruding five times from the Tip of a single Bone Armature in Edit Mode. The second chain as been created by Subdividing a Scaled up single Bone Armature (in Edit Mode) five times (Number of Cuts 5).

Either method of generating the chain is acceptable **BUT,** make note of the automatic names that have been displayed. Both chains have the lower Bone named: **Bone,** while the remainder of Bones are named in reverse order. **The Bone names are important.**



Properties Editor, Object Data Properties
(for the selected Armature Chain)



Extrusion **Subdivision**

Figure 22.50

For the demonstration of the Inverse Kinematic Constraint the **Extruded Chain** will be used.

The first step in the process is to create a **Control Bone.** When the Inverse Kinematic Constraint is applied to the Armature Chain, selecting and Translating the Control Bone will see the Bones in the chain follow.

To create a **Control Bone** the uppermost Bone in the Chain (Bone.005) will be disconnected from the Chain.

In Figure 20.51 the top Bone (Bone.005) has been selected in Edit Mode and disconnected from the Armature by pressing **Alt + P Key** and selecting **Clear Parent** in the menu that displays. In Edit Mode **Control** (the top Bone) has been moved aside.

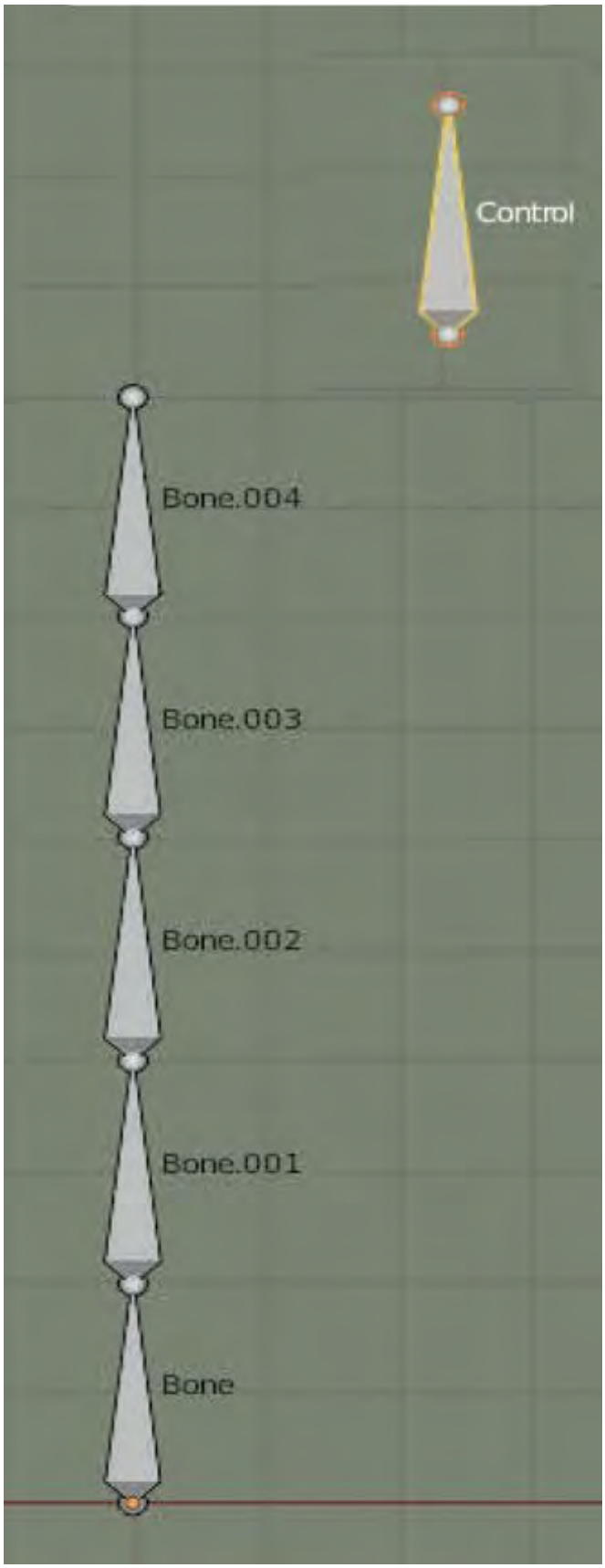

Control
Bone.004
Bone.003
Bone.002
Bone.001
Bone

Figure 22.51

Rename Bone.005 as **Control** in the Outliner Editor. Note that it is removed from the hierarchy of the Chain.

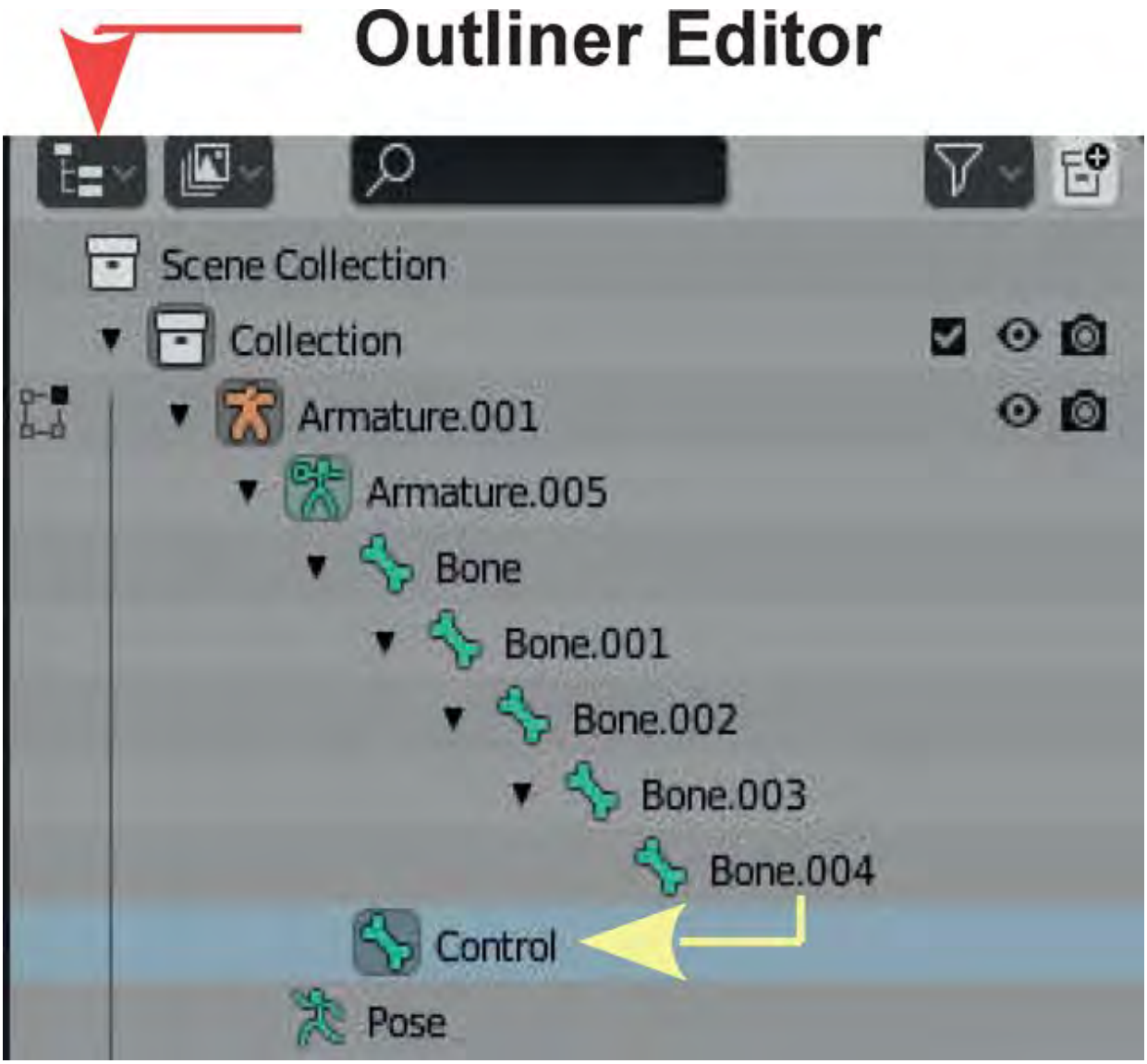


Figure 22.52

Renaming in the Outliner automatically renames the Bone in the 3D Viewport Editor.

The Bone named **Control** will be used for manipulating the Armature Chain.

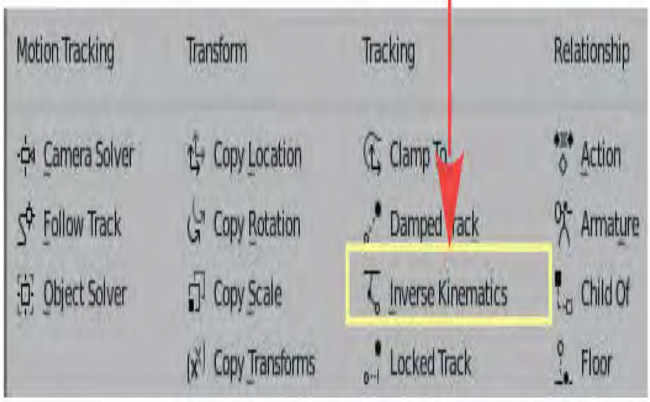
The Bones in the Armature Chain are connected, therefore, at this stage, an individual Bone in Pose Mode can only be Rotated about its Base or Scaled . Bones above in the chain will also Rotate or be Scaled but that is the limit of the Posing. Grabbing the Control Bone and moving it has no affect on the Chain.

To which Bone in the Chain the Constraint is applied determines how the Control affects the Chain. This is where Bone naming becomes important. If the Chain were being used to Pose parts of a Model it would be essential that the correct Bone was selected.

To use the Control an **Inverse Kinematic Constraint** has to be applied to **one of the Bones** in the Armature Chain.

Properties Editor

To demonstrate the IK Constraint select **Bone.004** in the Armature in **Pose Mode**. In the Properties Editor click the **Bone Constraints Properties** and click **Add Bone Constraint**. Select **Inverse Kinematics** in the menu that displays.



Bone Constraint Properties

Object Constraint Properties

Figure 22.53

In the IK Bone Constraint Panel (Figure 20.53) set **Target** as **Armature** and **Bone** as **Control.**

With the Constraint applied to Bone.004 you immediately see the Bone point towards the Control (Figure 22.54). Selecting **Control** and Translating sees Bone.004 being Translated with the Bone Chain following (Figure 22.55).

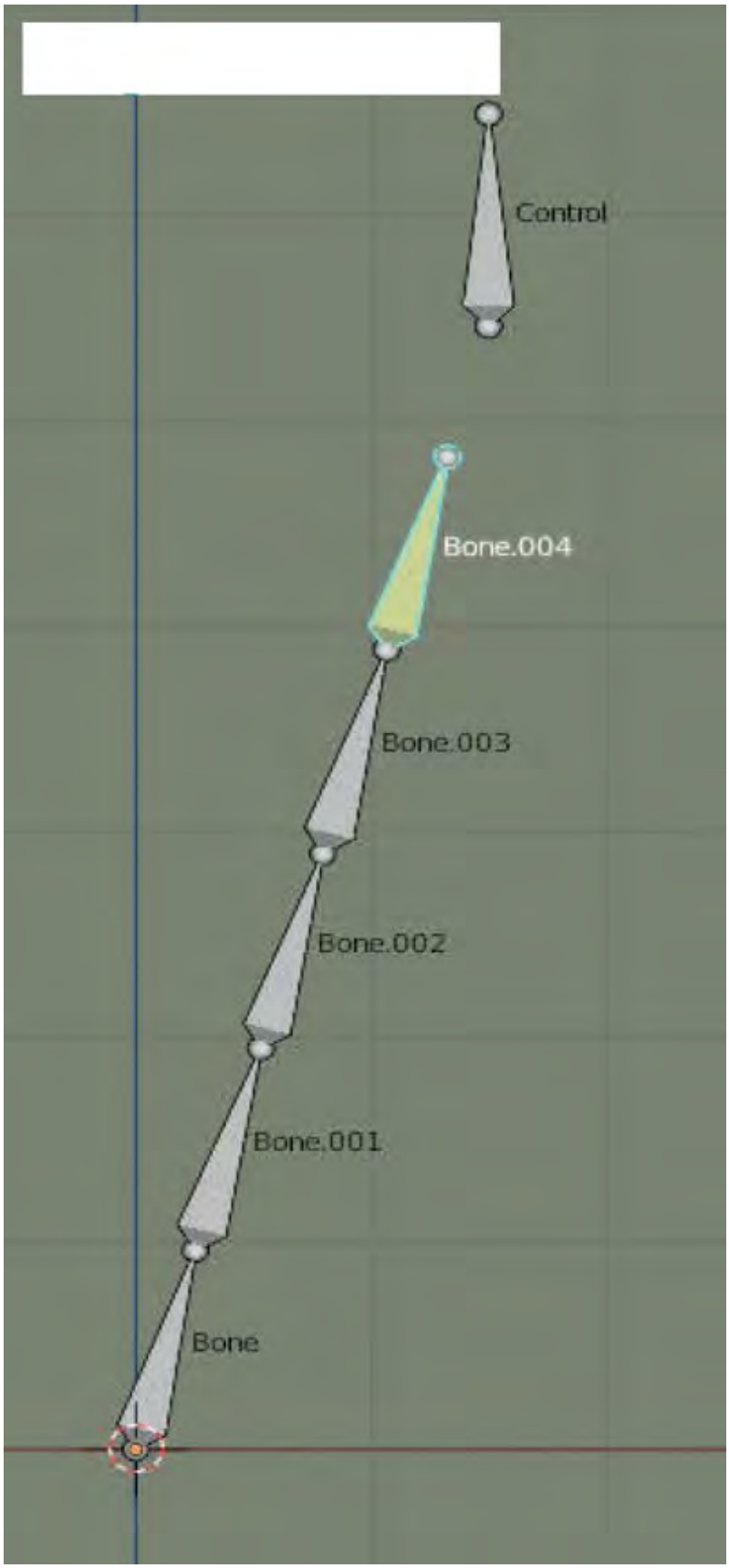

Control
Bone.004
Bone.003
Bone.002
Bone.001
Bone

Figure 22.54

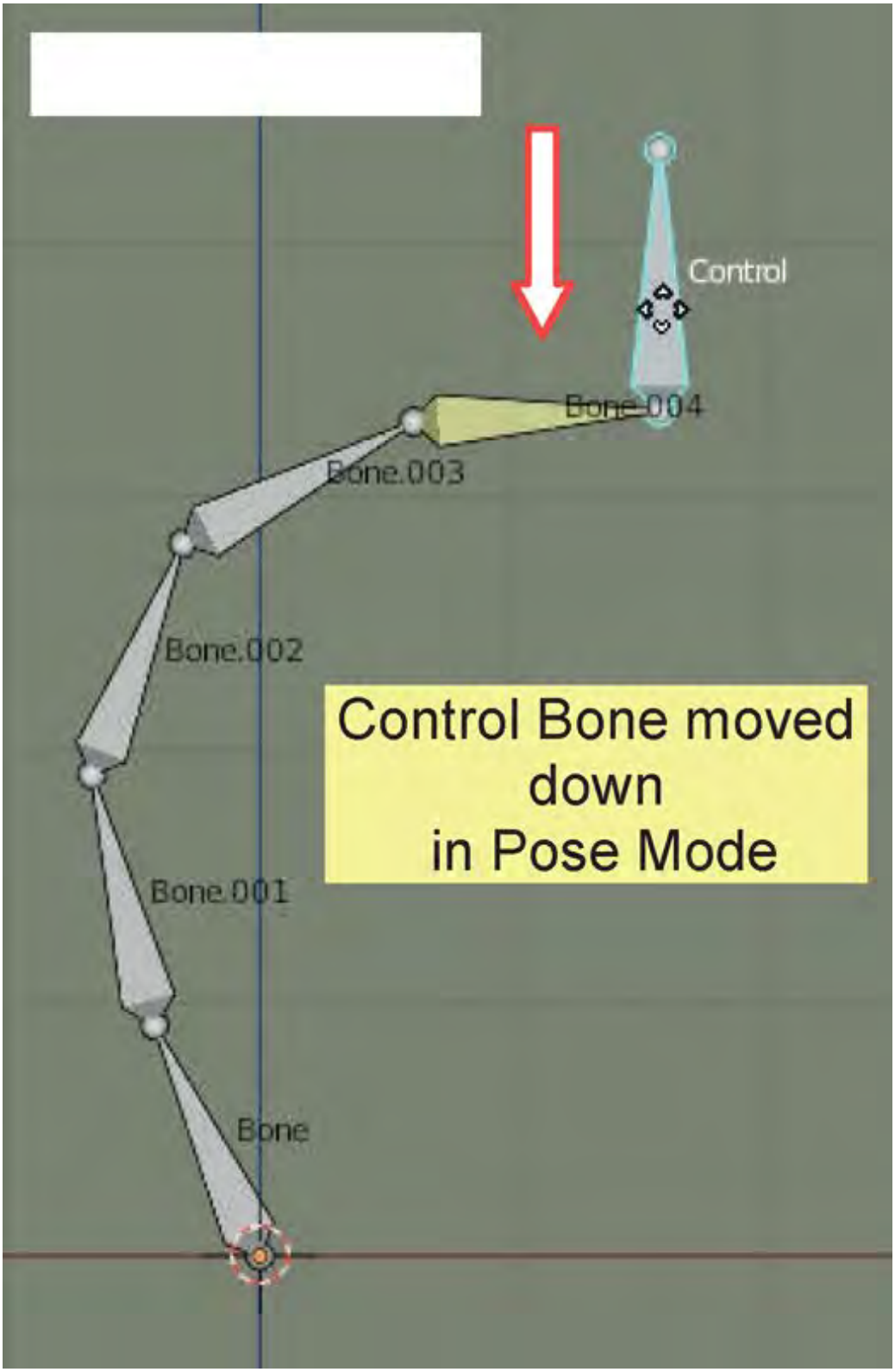

Control
Bone.004
Bone.003
Bone.002
Bone.001
Bone
Control Bone moved down in Pose Mode

Figure 22.55

Make particular note of the **Chain Length** setting in the IK Constraint Panel. Chain Length: 0 means all the Bones in the Armature Chain are affected.

Remember: The IK Constraint is applied to Bone.004.

By selecting Bone.004 (in Pose Mode) you see a broken line connecting the Tip of Bone.004 to the base of **Bone** (Figure 22.56) **NOTE:** This is when the Chain Length is set to 5 or 0. Changing the Chain Length sees the connection change (Figures 22.57 and 22.58)

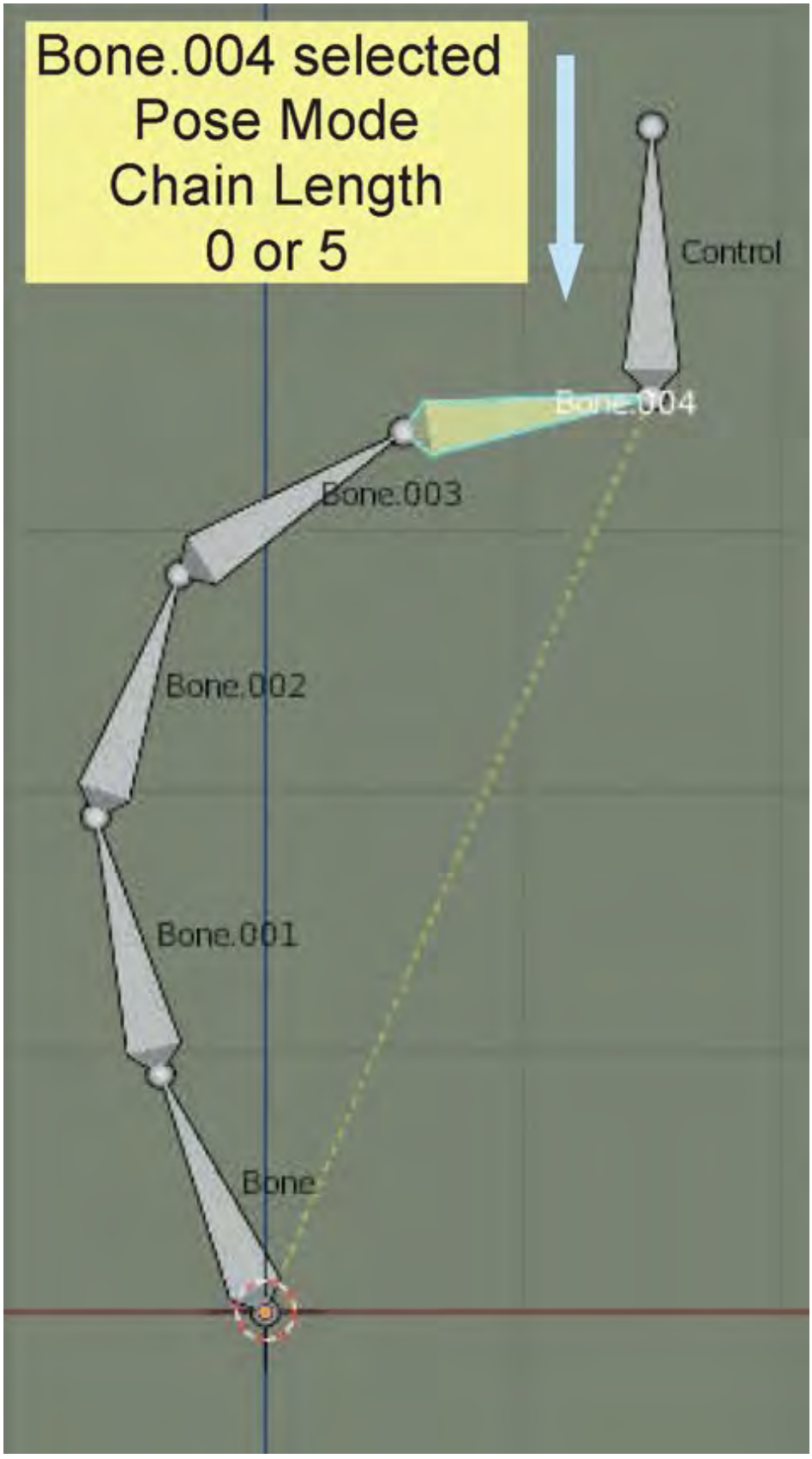

Bone.004 selected
Pose Mode
Chain Length
0 or 5
Control
Bone.004
Bone.003
Bone.002
Bone.001
Bone

Figure 22.56

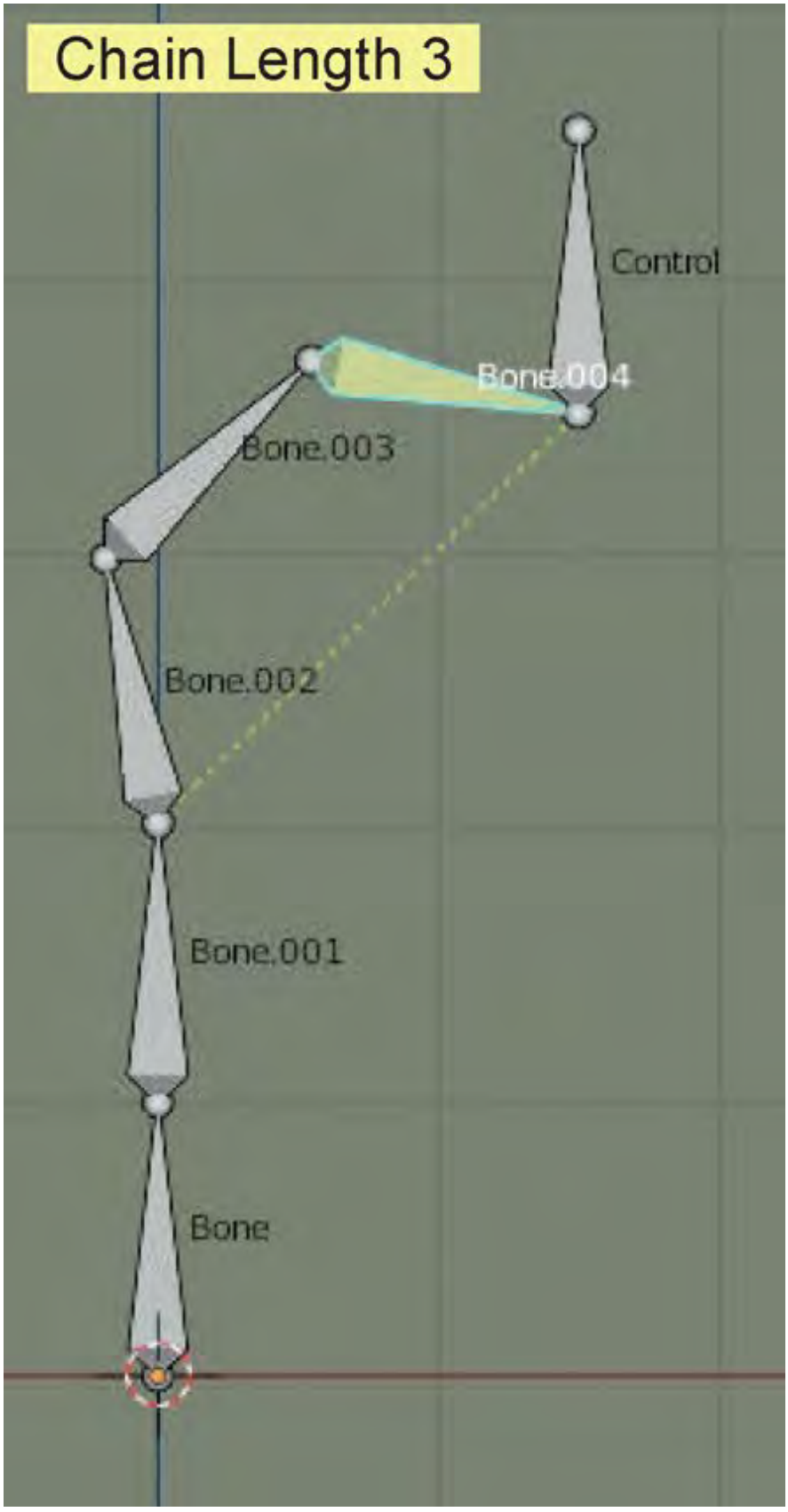

Chain Length 3
Control
Bone.004
Bone.003
Bone.002
Bone.001
Bone

Figure 22.57

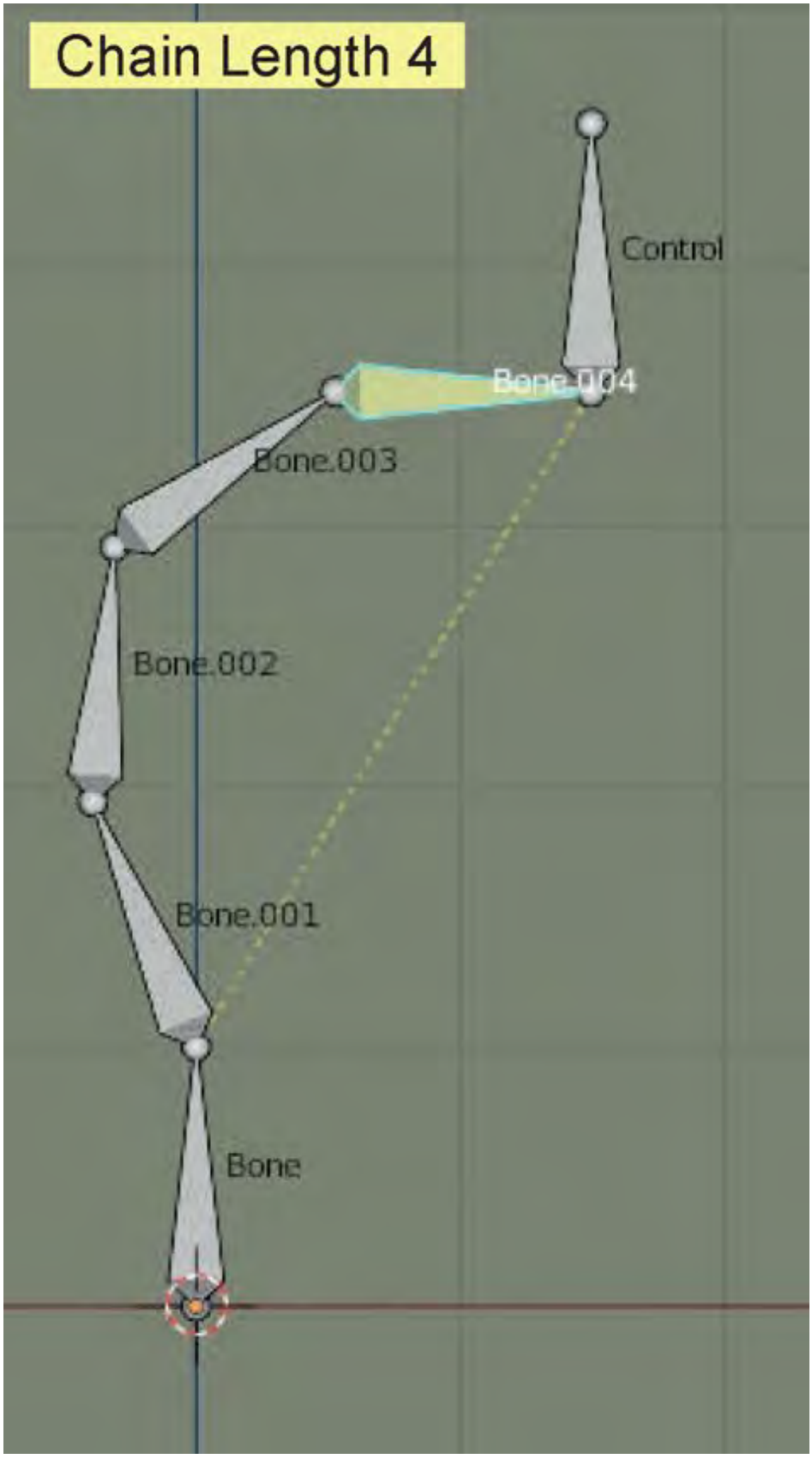

Chain Length 4
Control
Bone.004
Bone.003
Bone.002
Bone.001
Bone

Figure 22.58

The Chain Length connection determines how the Control Bone affects the Bone Chain.

22.16 Spline IK Constraint

The **Spline IK Constraint** forces a multi Bone **Armature** to follow the shape of a Curve. With the Armature constrained to the Curve, the Curve is then manipulated to adjust the shape of the Armature and in turn any mesh assigned to the Armature.

To demonstrate, in the default 3D Viewport Editor, delete the **Cube** and add a **Bezier Curve.** Scale the Curve up. Create a multi Bone **Armature** (Figure 20.59).

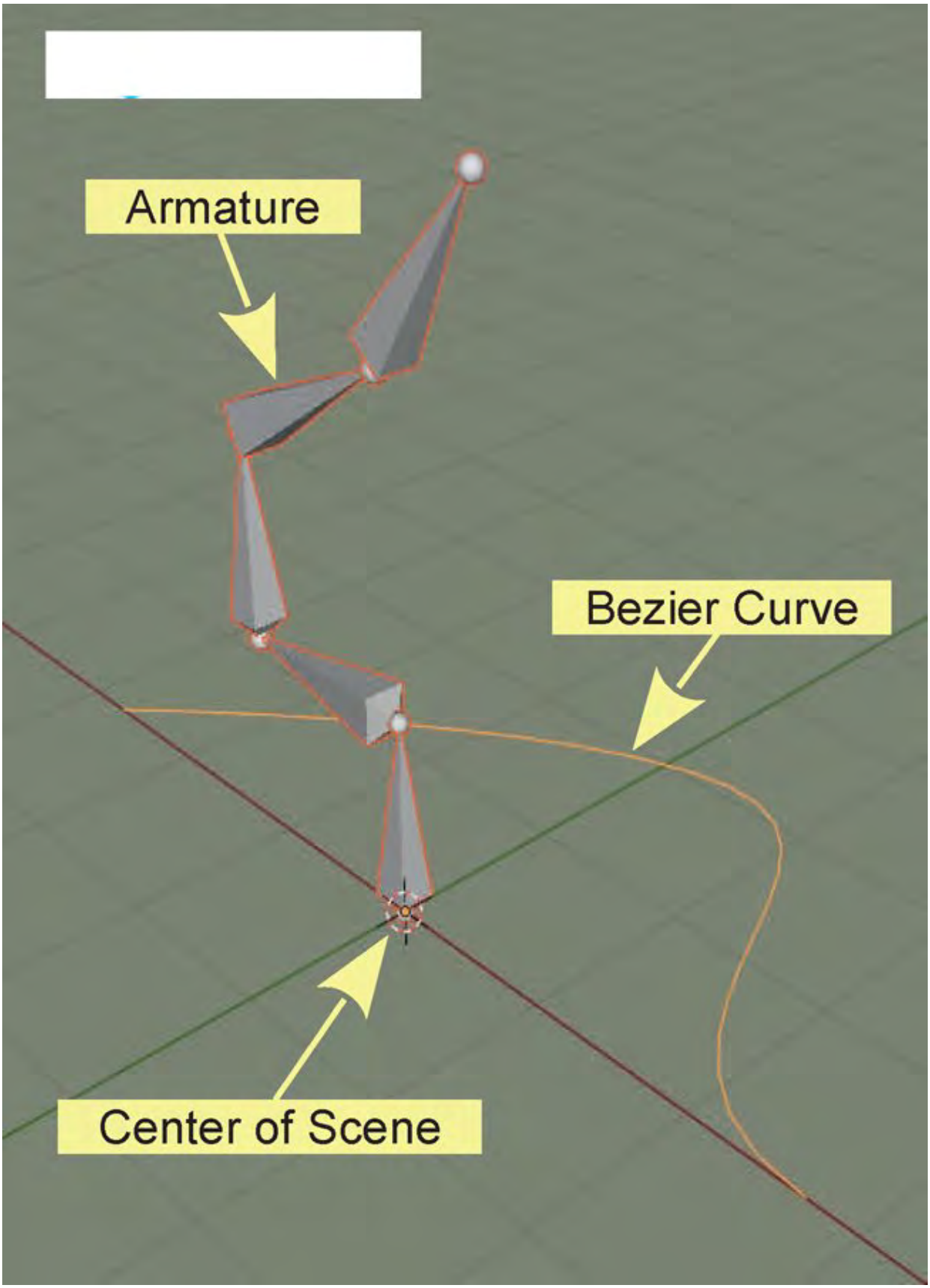

Armature
Bezier Curve
Center of Scene

Figure 22.59

Leave the 3D Viewport in **User Perspective View.**

Both the of the Armature and the center of the Bezier Curve are located at the center of the Scene.

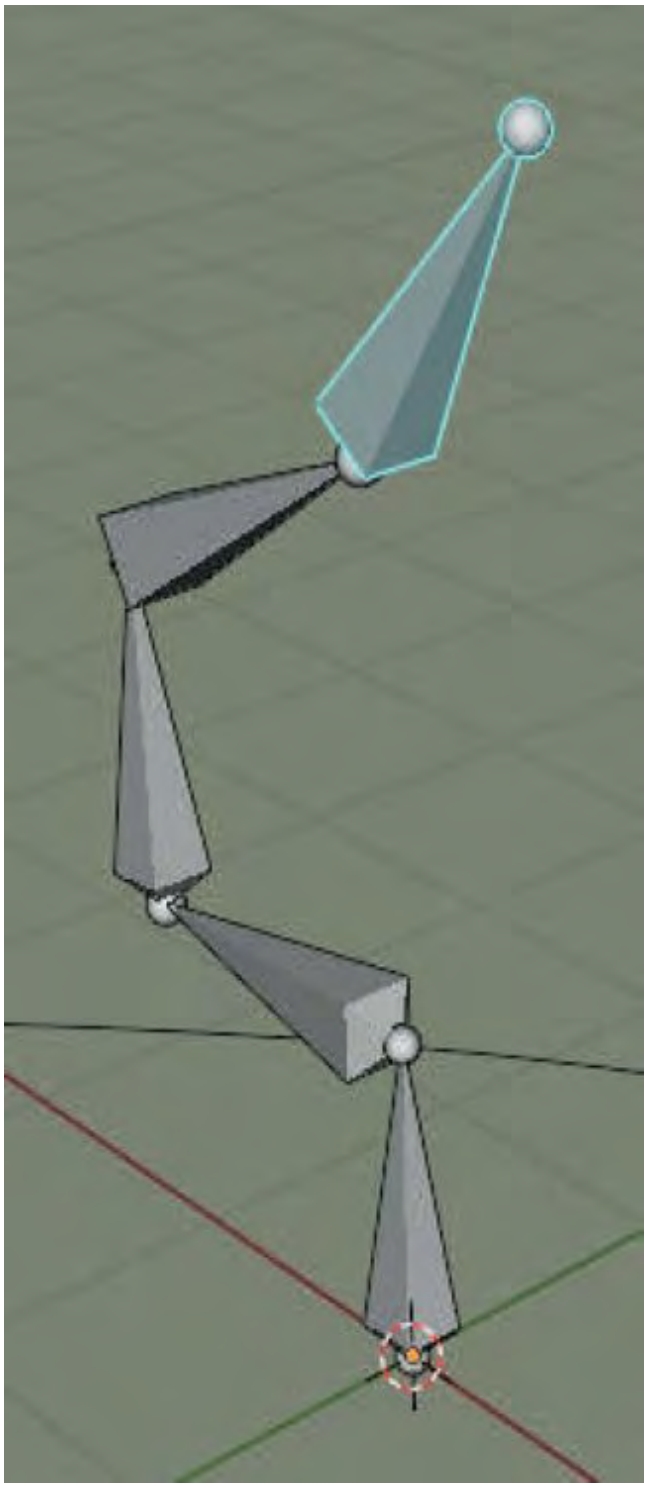


Figure 22.60

With the Armature selected, go into **Pose Mode. When selected,** the outline of the Armature will be displayed in blue.

Deselect the Armature then select the top Bone in the Armature (Figure 20.60).

In the **Properties Editor, Bone Constraints Properties** add a **Spline IK Constraint** (Figure 20.61). Change the **Spline Fittings: Chain Length** value to 5 (the number of Bones in the armature). In the **Target** panel select **Bezier Curve.**

Motion Tracking
Transform
Tracking
Relationship
Camera Solver
Follow Track
Object Solver
Copy Location
Copy Rotation
Copy Scale
Copy Transforms
Limit Distance
Limit Location
Clamp To
Damped Track
Inverse Kinematics
Locked Track
Spline IK
Stretch To
Action
Armature
Child Of
Floor
Follow Path
Pivot
Armature
Bone.004
Add Bone Constraint
Spline IK
Target
BezierCurve
Influence
1.000
Fitting
Chain Length
5
Even Divisions
Chain Offset
Chain Scaling
Use Curve Radius
Y Scale Mode
Fit Curve
XZ Scale Mode
None

Figure 22.61

The Armature is relocated in the 3D Viewport Editor and shaped to the Curve.

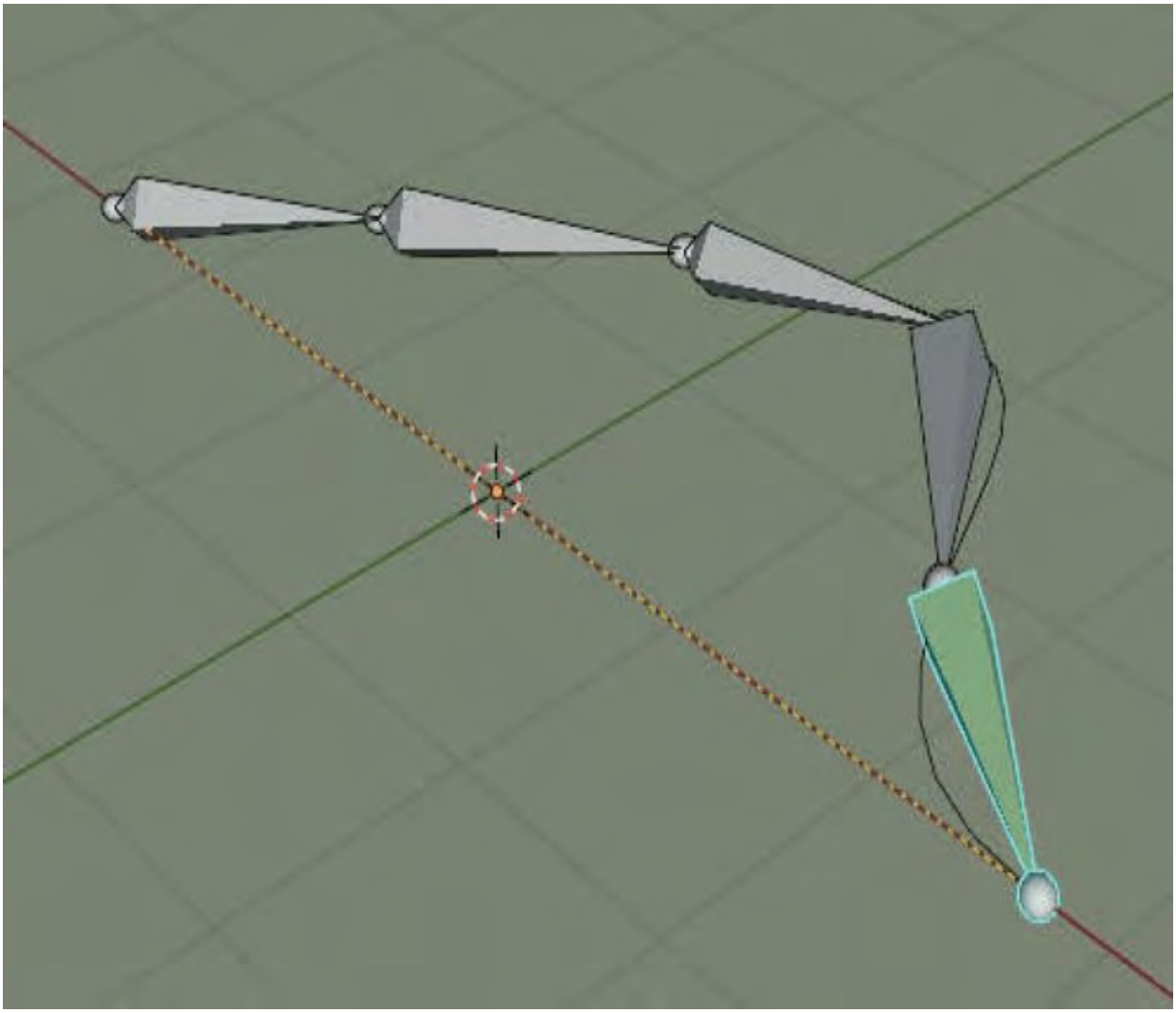


Figure 22.62

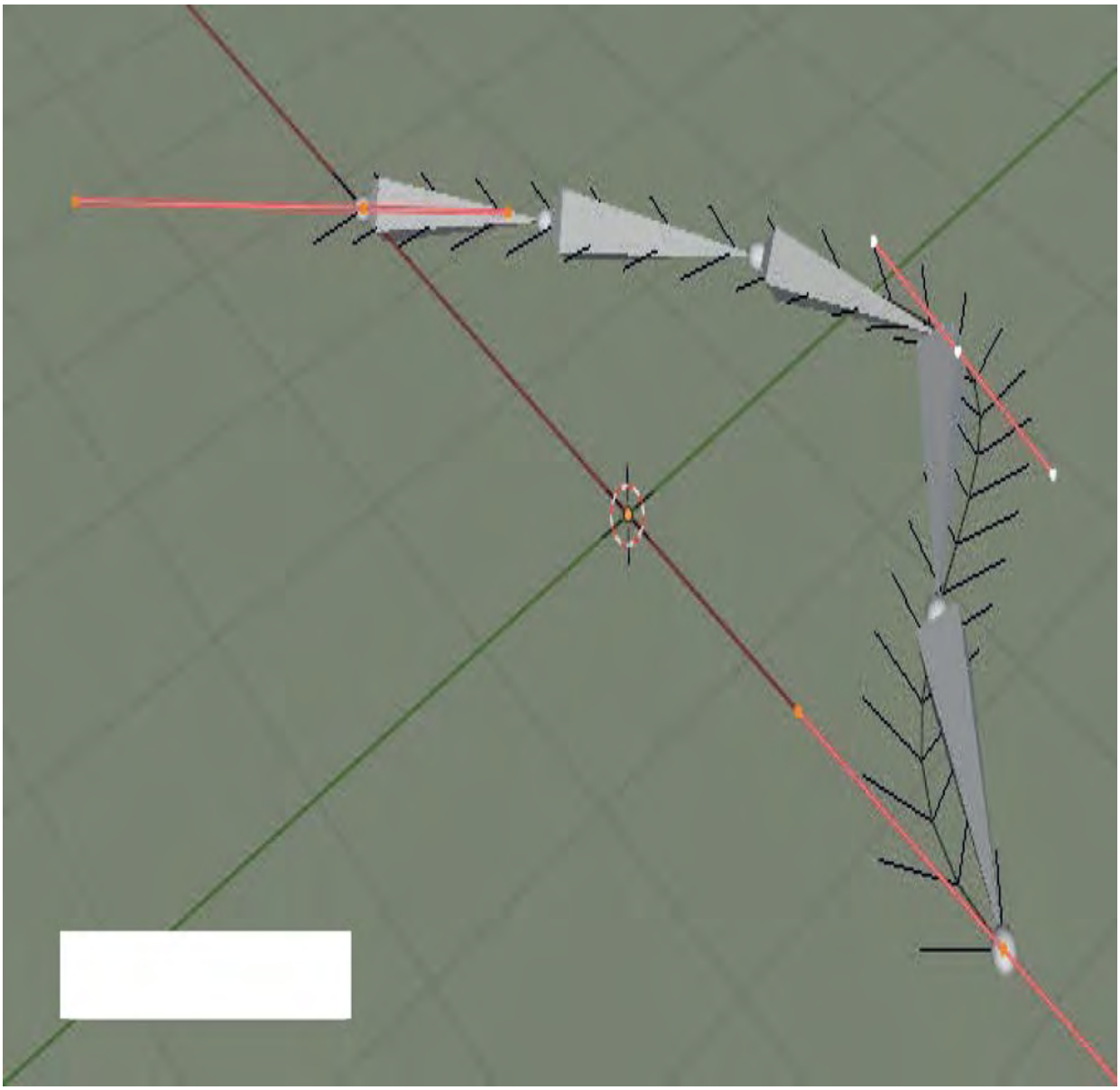


Figure 22.63

The Armature Bones are arranged with the direction of the curve. If the Curve were being used as an Animation Path, the movement along the path would be in the direction of the chevrons spaced along the Curve (Figure 20.62).

The direction may be reversed **in Edit Mode** by clicking RMB to display the **Curve Context Menu** and selecting **Switch Direction.** In doing this the chevrons are reversed and so are the Bones in the Armature.

With the Armature constrained to the curve the **Armature** may be posed by selecting the **Control Handles** on the **Bezier Curve.** The Curve may be subdivided in Edit Mode to add additional control handles and give more control over the posing. Remember the practical use of the Armature is to control the shape and movement of a mesh Object which is assigned to the Armature.

Hooks may be assigned to the control handles of the Curve giving a non renderable Object with which to translate and pose the Armature (Chapter 9 – 9.7). To add a **Hook** place the Curve in **Edit Mode** and ensure everything is **deselected.** Select a Control Handle then press **Ctrl + H key** and select **Hook to New Object** in the menu that displays. A **Hook** is displayed in the form of a **3D Cross.**

Anther method of introducing an non renderable Objects to allow Curve manipulation, when you have an Armature constrained to the Curve, is to add single Bones. You then parent the Bone to the Hook.

22.17 Forwara

With knowledge of Armatures and how they are constructed, manipulated and used to deform Mesh Objects you are placed to learn the intricacies of Character Rigging. This is the process of constructing an Armature to fit a Character model. The model can be anything your imagination allows but for demonstration purposes, the subject will be devoted to Character Rigging a Humanoid Figure.

A preliminary introduction to **Character Rigging** has been presented in section 22.8 **Building an Armature Rig.** Since this topic is such an important part of Modeling and Animation it is felt that a more detailed study is required. This is presented in Section 22.18 to follow.

There are pre-assembled Rigs you may download from the internet which are free to use, but using a ready made Rig for your particular application may

require you to modify the download. It is therefore advisable to understand how a Rig is constructed from basics.

22.18 Character Rigging

As you have seen Character Rigging is the process of creating an Armature to suit a Model of a Character then associating the Bones in the Armature with parts of the Model. The Bones are Posed and animated and the parts of the Model follow suit. To facilitate Posing, special Control Bones are incorporated in the Armature to manipulate the Bones.

To expand and consolidate this topic an exercise in Rigging a cartoon character will be presented which demonstrates constructing an Armature for a pre-constructed Model.

To constructing an Armature and Rigging you require a Mesh Model of a Character.

Free to use models may be downloaded from the internet or you may construct your own model in Blender or use an external application such as the Make Human program.

Constructing a crude humanoid figure by extrusion in Blender was discussed in Chapter 11-11.6 and the basics of building a Rig for a human Model has been demonstrated in section 22.8 of this chapter.

Constructing a model of a human figure can be a lengthy process, depending on the detail employed. There are several websites where you can download pre-built models, some of which are pre-rigged. To understand the rigging process you should begin with a simple **Low Poly Mesh** model. **Low Poly** means a mesh model with a minimum number of Vertices, Edges and Faces.

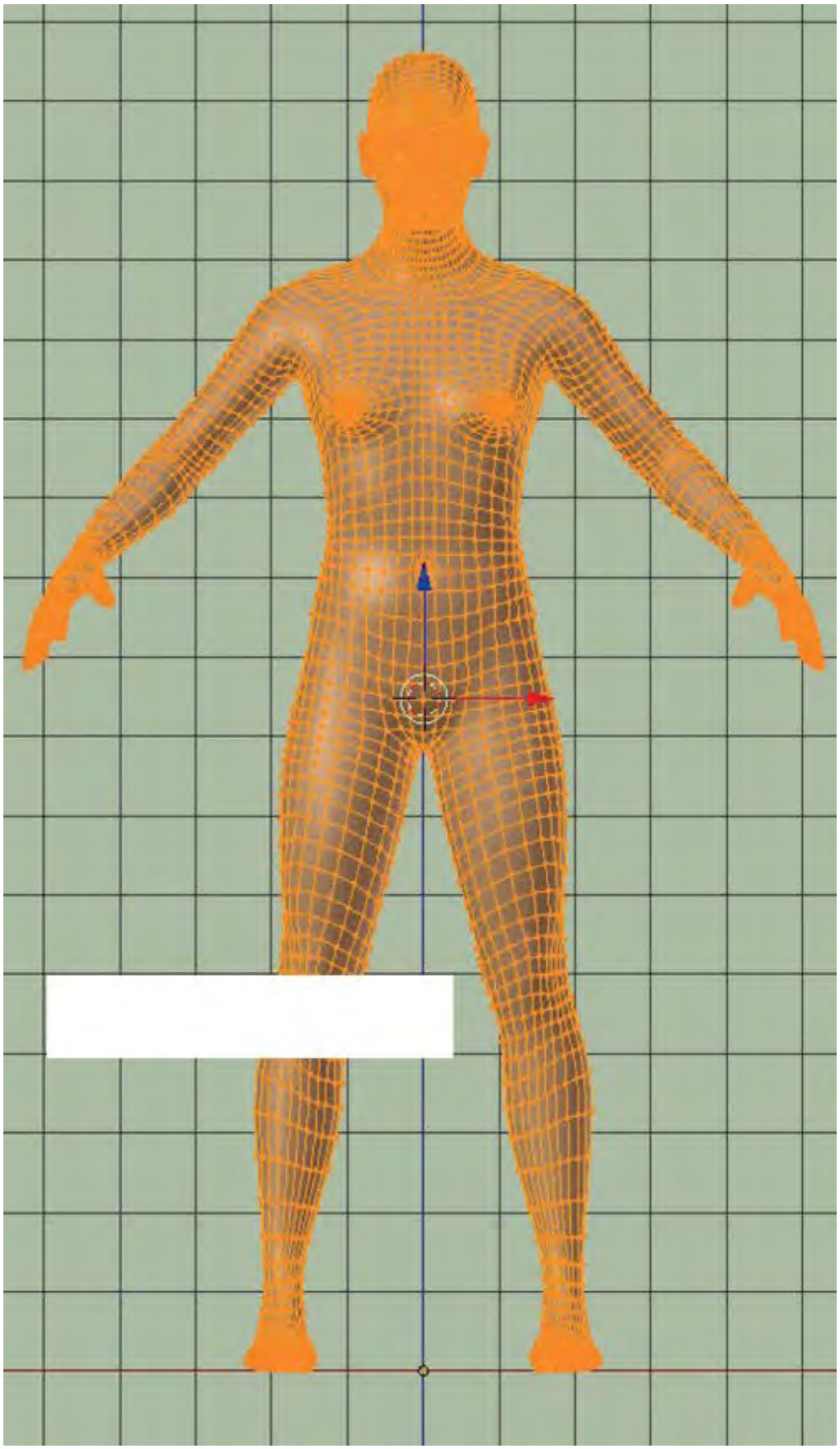


Figure 22.64

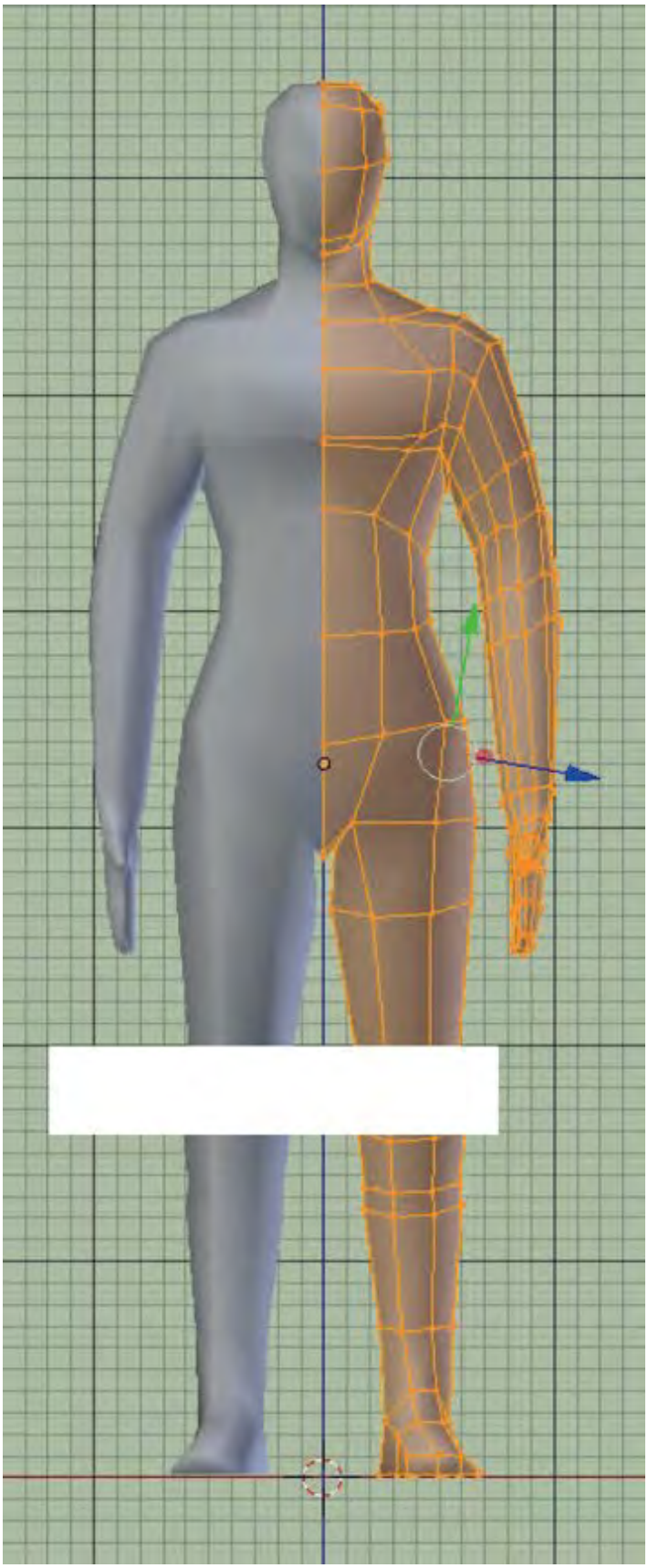


Figure 22.65

Figure 22.64 shows a model created using the **Make Human** program. As you see there is a vast difference in the number of vertices in the **Low Poly** model in Figure 22.65.

The **Low Poly** Model of a human as shown in Figure 22.66, supplied by **tweediez,** released under the Creative Commons Attribution 3.0.

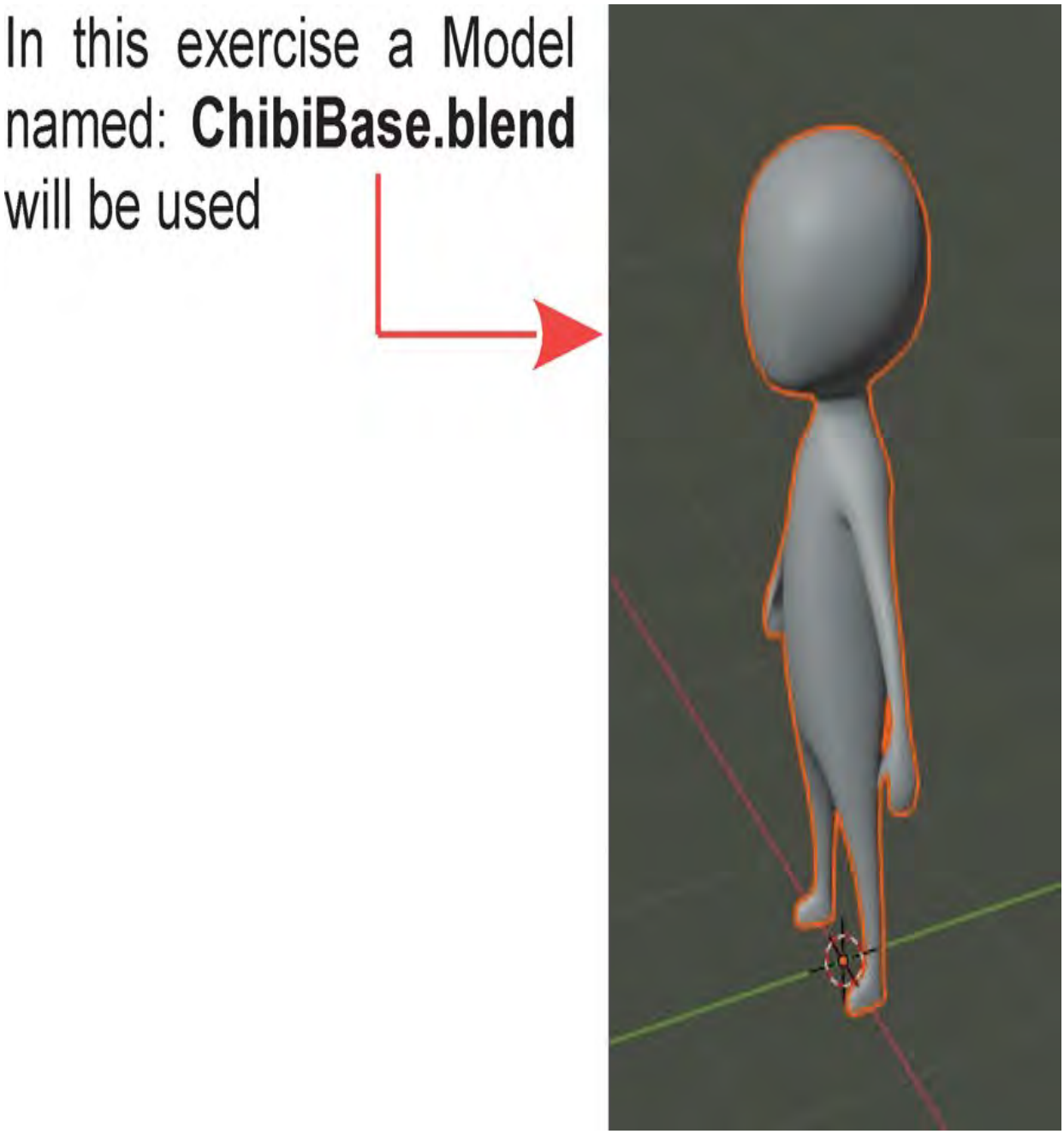


Figure 22.66

Rigging a Character Model is a reasonably intricate operation. This exercise is intended as an introduction which will allow you to understand detailed tutorials. The Rigging process, even at a basic level, requires patience, perseverance and attention to detail.

Before downloading the Model look at the following figures. They show examples of an Armature Rig and a pre-rigged Model.

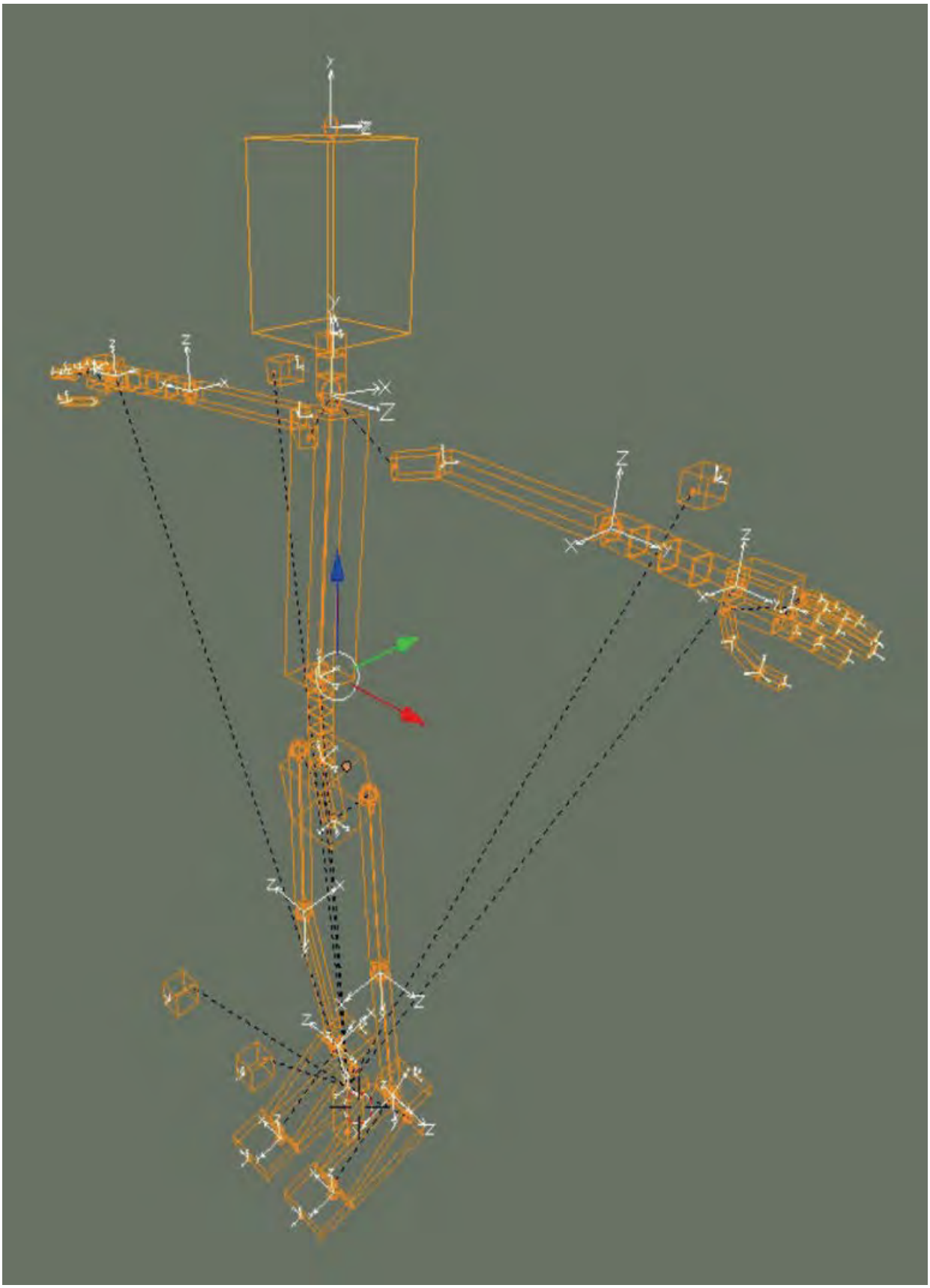


Figure 22.67

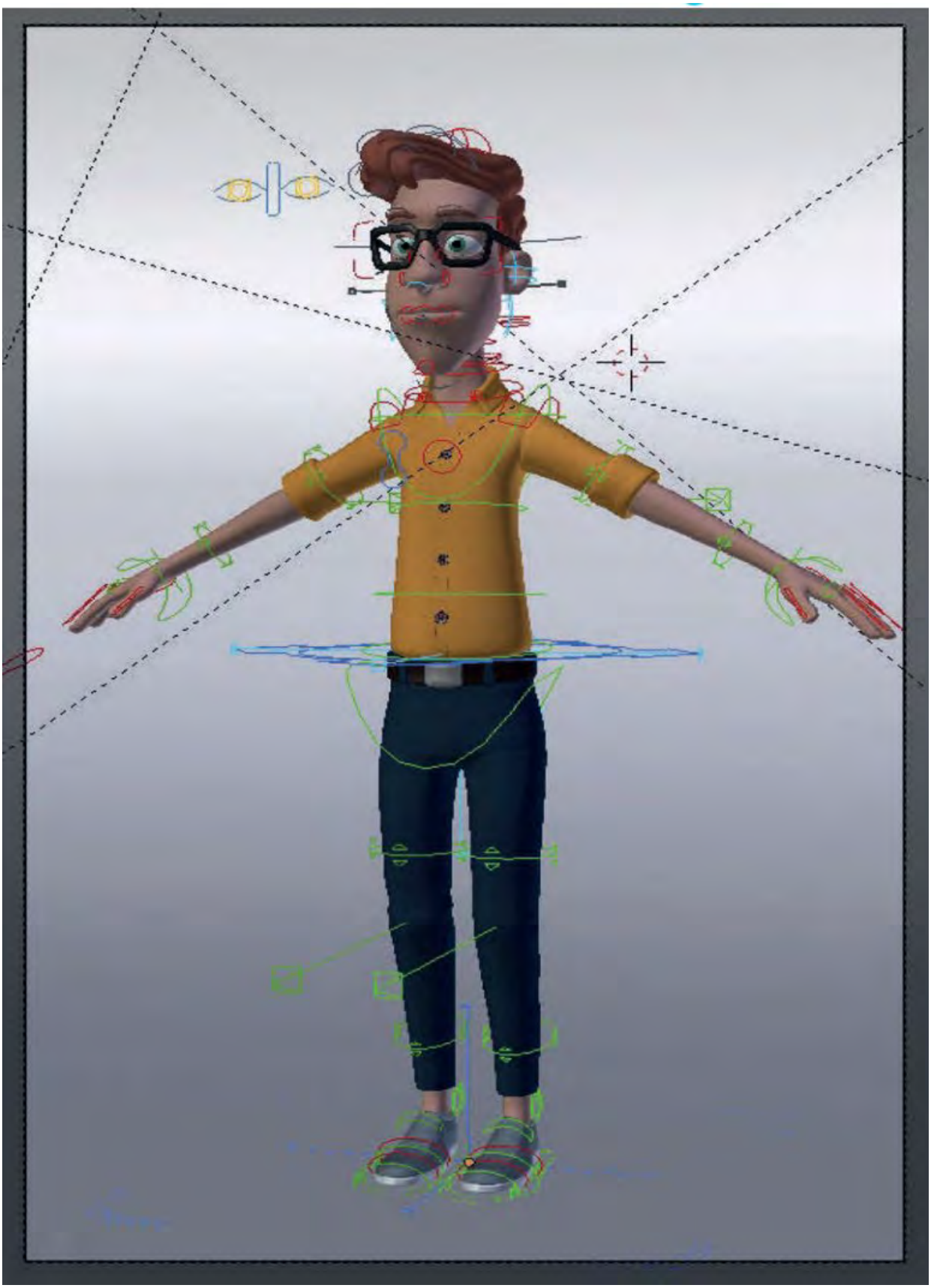


Figure 22.68

Figure 22.67 is a reasonably complex Armature Rig with the Viewport Display Type: **B-Bone.** The 3D Viewport Editor is in Solid Viewport Shading Mode with **Toggle X-Ray** engaged.

Figure 22.67 shows a fully rigged Model with Armature Controls made from various shaped Objects. This demonstrates that Armature Rigs are not limited to the Armature Display Types.

Download a Character Model

The Character Model to be used is named **Chibi.** To get this model go to the **Blend Swap** Website;

http://www.blendswap.com/

To download a model you have to register as a member. This means entering an email address and creating a user password and agreeing to the terms of use. If you don't do this you won't be able to download any of the fantastic **free models** available on the site.

Assuming that you have signed up, log in, then click on **Search** in the header at the top of the website home page. Select **Search** in **Blends** and enter **Chibi** in the Search Keywords bar. Click on **Search.**

The Model you are looking for is titled **CHIBI MODEL BASE.**

When you are logged in there will be a download link. The download is a ZIP file which you have to decompress to get the Blender file containing the model. Remember where you download to, and where you unzip to on your computer.

At this point it is assumed you have the file downloaded and unzipped.

The unzipped file is named **ChibiBase.blend** and will be located on your computer where you unzipped to.

Important: Although **ChibeBase.blend** is a Blender file with the **.blend** suffix and it will open in Blender, you may find anomalies when opening the file due to the version of Blender in which it was created. That being the case and since you are only seeking to obtain a model on which to work, it is better to Append the Mesh Model from the ChibiBase.blend file into a **new Blender file** opened in your version of Blender.

How to Append an element from one Blender file to another is described in Chapter 3 – 3.12.

To save you the frustration of reviewing that procedure, open a new Blender file. Delete the Cube Object then click on **File** in the Screen Header and click Append in the menu that displays. The File Browser Editor opens where you go find your ChibiBase.blend file.

Double Click on the file name in the File Browser to see the content of the file. Double Click on **Object** in the menu.

The Chibi model has probably been developed by modeling from the default Cube Object which appears in a new Blender file and the name hasn't been changed. In the Object folder **Chibi** is, therefore, seen as **Cube.**

Click on **Cube** in the folder menu, then click on **Append.**

Say hello to Chibi

Figure 22.69

Chibi is a humanoid character (Figure 22.68). Use your imagination to decide whether Chibi is a child or an alien or a little bow legged guy with a big head. Chibi can be whatever you decide.

With Chibi appended save the new Blender file a new name (Chibi.blend).

To proceed, it is assumed you have the new Chibi.blend file opened. Chibi displays with a red outline and there are no Object Properties in the Properties Editor. Click in the 3D Viewport Editor to deselect Chibi then click on the Chibi Model to select it. The outline will turn orange and Object Properties will display in the Properties Editor.

You are good to go. **Have the 3D Viewport Editor in Front Orthographic View.**

Note: Chibi is a Mesh Model. . You may see a slight resemblance to the model created in Chapter 11–11.6. Chibi is a much nicer model.

Chibe has been created by **Magiclass** and released under the **CC Zero License.**

Note: When you Append a Mesh Model from one Blender file to another you also Append the Model's properties, such as, any Modifiers used in construction.

Prepare the Character Model

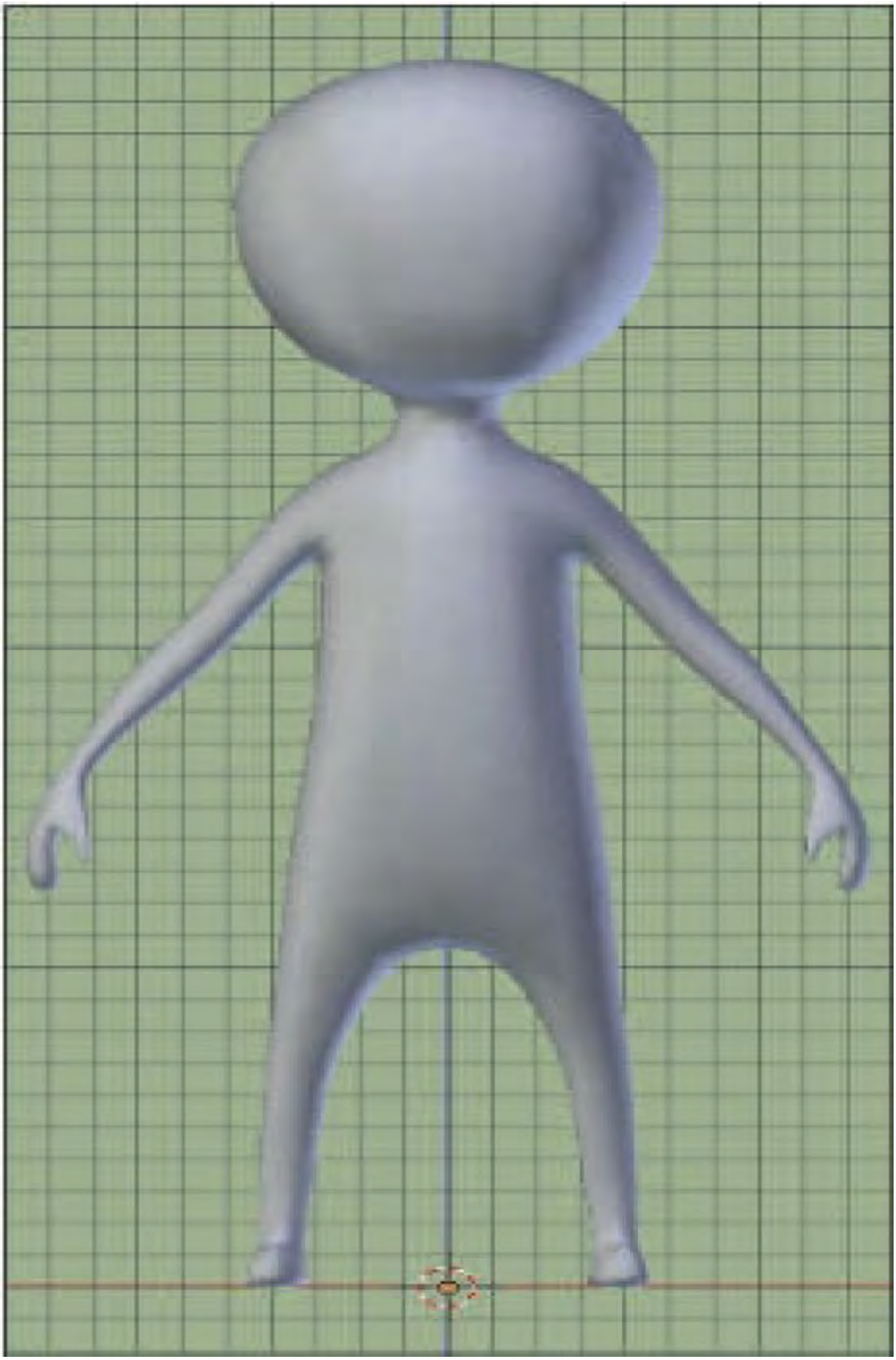


Figure 22.70

When you download a model or append from another Blender file, you should examine it before using. In examining **Chibi** you will see, in the **Properties Editor, Object Data Properties, Vertex Groups Tab,** that **Vertex Groups** have been generated. These are probably left over from a previous Blender operation. To begin with a clean slate, click on the minus button at the side of the list until **all groups** have been deleted.

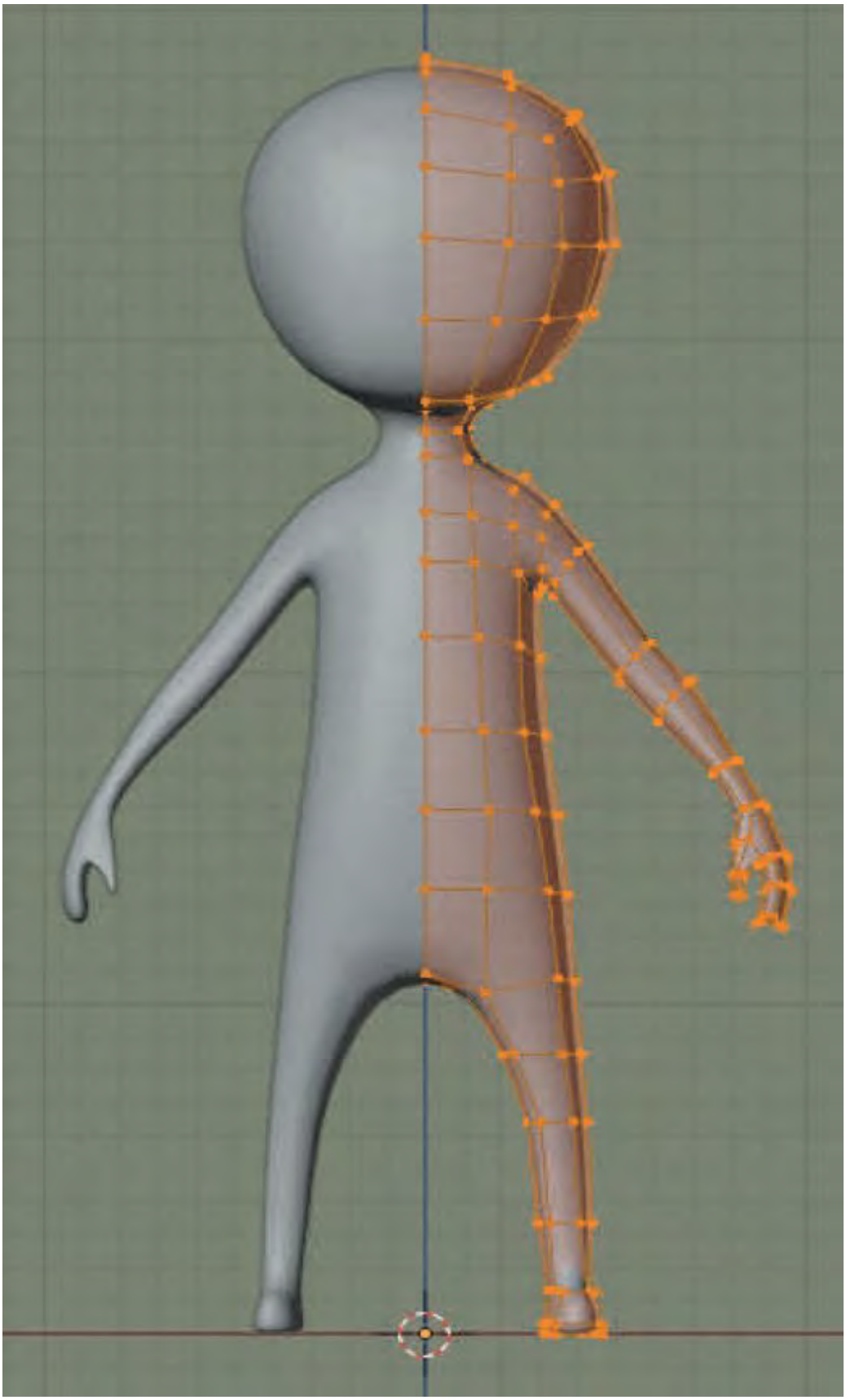


Figure 22.71

Cube
Add Modifier
Mirror
Axis
X
Y
Z
Bisect
X
Y
Z
Flip
X
Y
Z
Mirror Object
Clipping
Merge
0.001 m
Bisect Distance
0.001 m
Data
Subsurf
Catmull-Clark
Simple
Levels Viewport
1
Render
2
Optimal Display
Advanced

Figure 22.72

Take a closer look at Chibi. **Tab** into **Edit Mode.** You see that the model only has Vertices on the right hand side (your right, Chibie's left) (Figure 22.71). In the **Properties Editor, Modifier Properties** (Figure 22.72) you will see that a **Mirror Modifier** and a **Subserf** (Subdivision Surface) **Modifier** have been used when creating the model. The Mirror Modifier allows Vertex creation and manipulation on one side to be mirrored on the other side. Subserf makes the surface of the model appear smooth without increasing the Vertex count.

Add More Vertices

Actually, it will be better to have a few more Vertices when posing the figure go into **Object Mode** and click **Apply** in the **Subserf Modifier** panel **(in Object Mode).** You have to be in Object Mode to Apply a Modifier. Note: Before applying the Modifier that, in the Modifier panel, the **Levels Viewport value is 1** (Figure 22.73). This means that when the Modifier is the mesh surface of the model will be subdivided once. In other words the Vertices will be doubled. In Edit Mode you will see there are more Vertices than before. You also want the model to have Vertices on both sides, so go back into **Object Mode** and apply the **Mirror Modifier** (click the Apply button) (Location Similar to Figure 22.73).

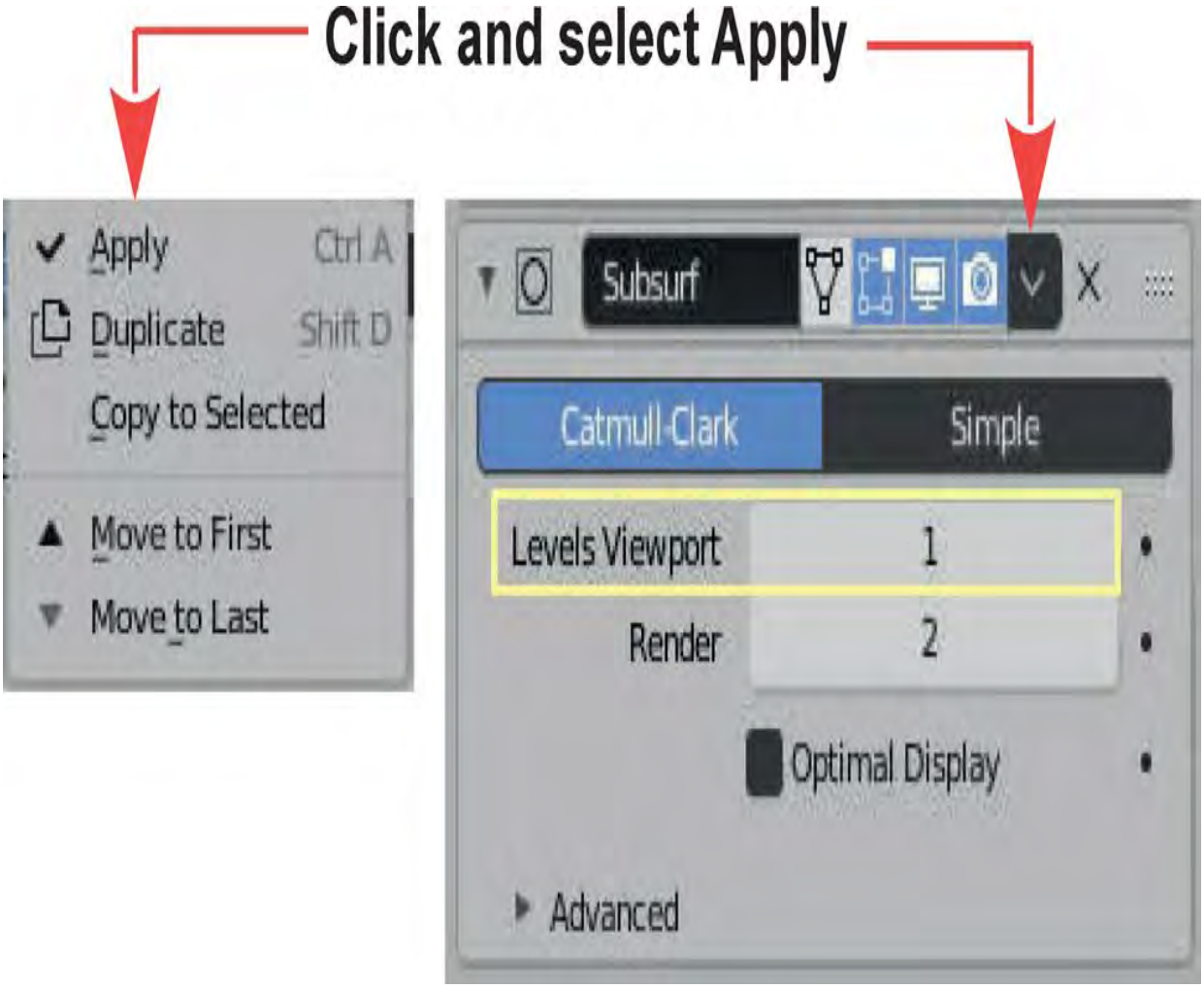

Click and select Apply
Apply Ctrl A
Duplicate Shift D
Copy to Selected
Move to First
Move to Last
Subsurf
Catmull-Clark
Simple
Levels Viewport 1
Render 2
Optimal Display
Advanced

Figure 22.73

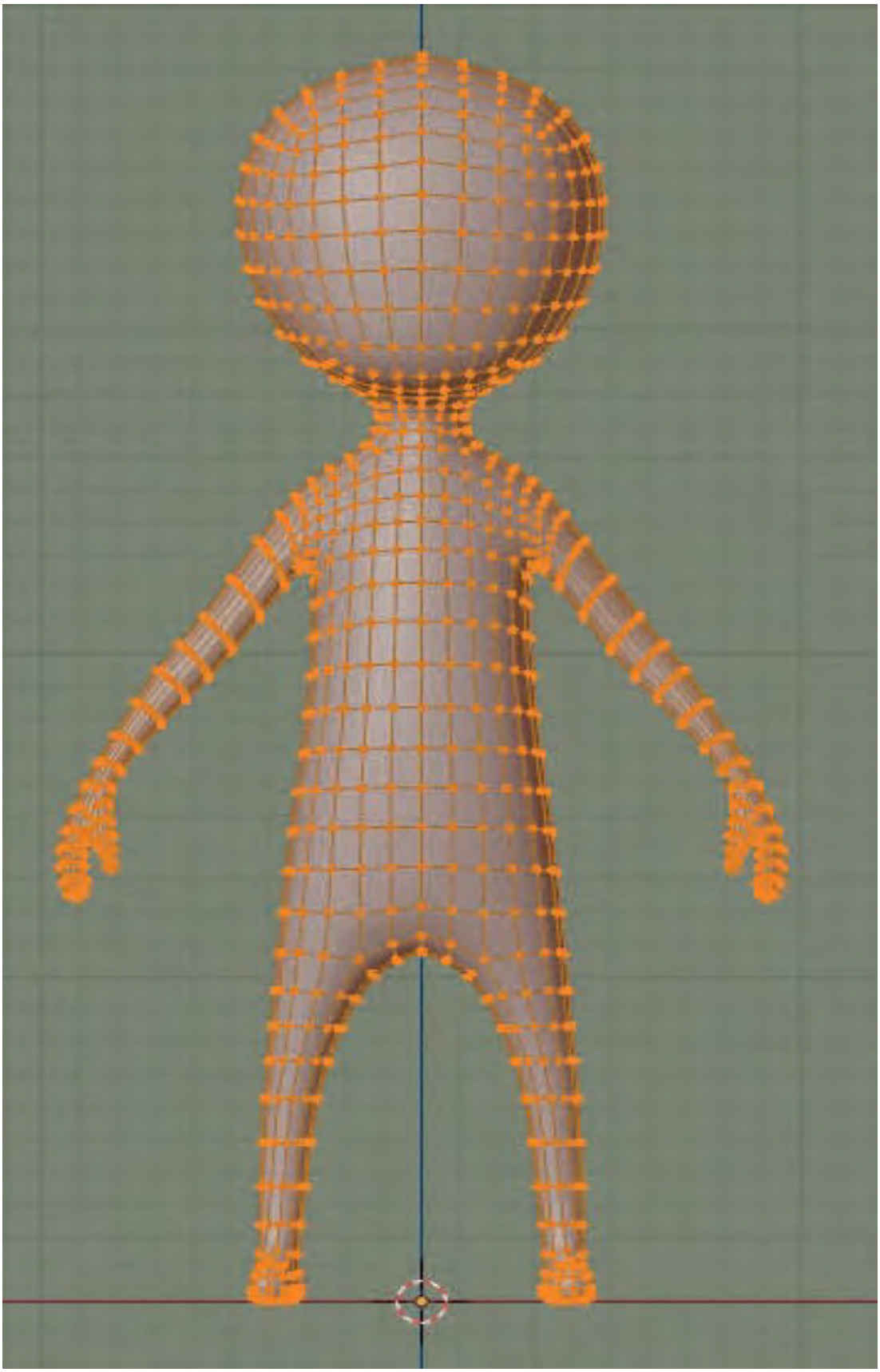


Figure 22.74

Figure 22.74 shows Chibi in Front Orthographic View with the Subserf Modifier and the Mirror Modifier applied.

Some Definitions

This chapter has been titled Character Rigging, therefore, what does Character Rigging mean?

A Character refers to a Model, in this case a model of a Humanoid Character. The model is a Mesh Model in that it is made up of a surface mesh constituting Vertices connected by Edges. There is nothing inside the mesh to begin with.

To pose the Mesh Model an Armature is constructed inside the mesh, specifically a Multi-bone Armature (single individual Bones connected together). The Armature may be considered as a skeleton for the Human Character.

Areas of the surface mesh are assigned (associated with or connected) to single Bones of the Armature or to multiple Bones.

Following the assignment of vertices to the bones the Bones will be manipulated (Translated or Rotated) to set the mesh in various poses or animated to move. When the bones move the surface mesh associated with the bones follow.

Remember: Armatures (Bones) are Non Renderable Objects, therefore, when an image or a video clip is rendered the Armature (Bones) are not included in the output.

Creating the Armature (Root Bone)

The Root Bone is the starting point for creating the Armature and will act as the primary Control Bone for moving and manipulating the Model in the Scene.

With Chibi displayed as shown in Figure 22.74, **Tab to Object Mode** in **Front Orthographic View.** This presents the figure face on with the Object's center located on the center of the Scene mid way between the feet. The center of the Scene on the XZ Axis is located between the feet.

Make sure you have the 3D Viewport Editor Cursor located at the center of the Scene. Be in Object Mode with **everything deselected**. Press **Shift + S Key** to display the **Snap Pie Menu** (Figure 22.75) and select **Cursor to World Origin**.

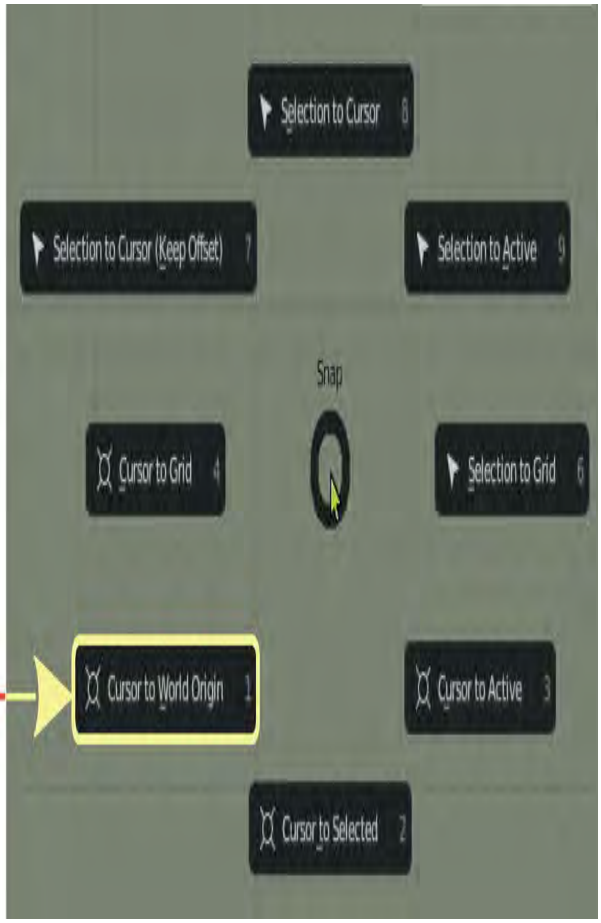


Figure 22.75

Tip: Save the Blender file repeatedly at each stage of the exercise. If you get off track it will be frustrating to have to repeat the entire •rocedure.

With Chibi **deselected** add an **Armature** (by default a Single Bone). The Bone is entered at the location of the 3D Viewport Editor Cursor at the center of the Scene (Center of World Origin). This will be the **Root Bone.** With the

Root Bone bone selected **Tab to Edit mode.** The Tip of the Bone will be selected as shown by the orange outline.

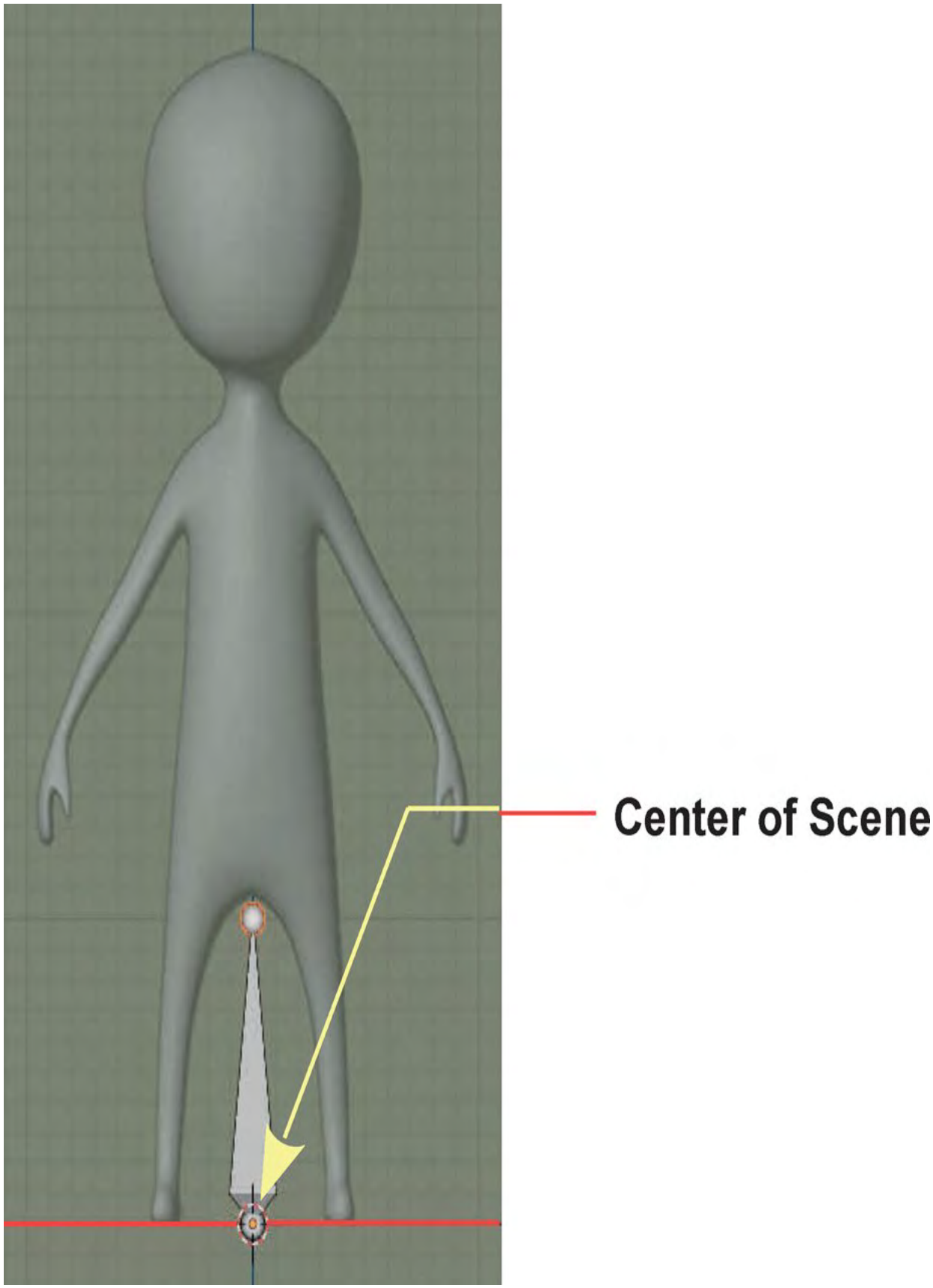

Center of Scene

Figure 22.76

The Bone is orientated vertically with its Tip at the top and with the Base of the Bone accurately located at the center of the Scene (Figure 22.75). Lay the Bone flat along the ground plane of the Scene in Right Orthographic View, on the YAxis (Figure 22.77).

To lay the bone flat and accurately, position it on the YAxis and use **Snap to Grid.** With the Tip selected in Edit Mode, **G Key** (grab), drag the Mouse down and locate the Tip on the Y Axis. To accurately position, click **Armature** in the Header, select **Snap – Selection to Grid.**

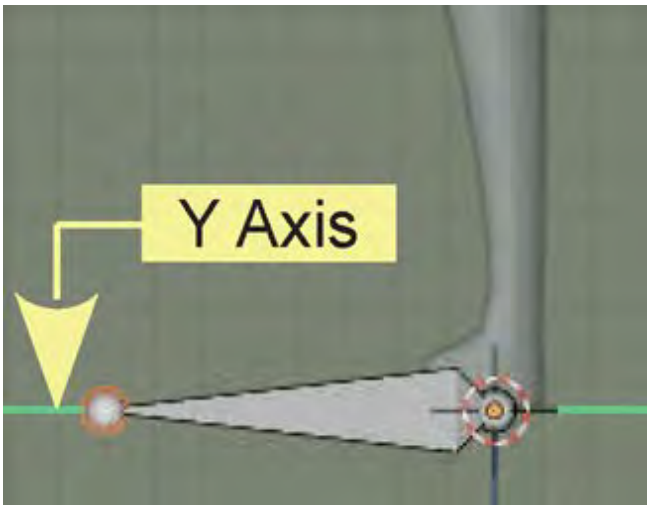
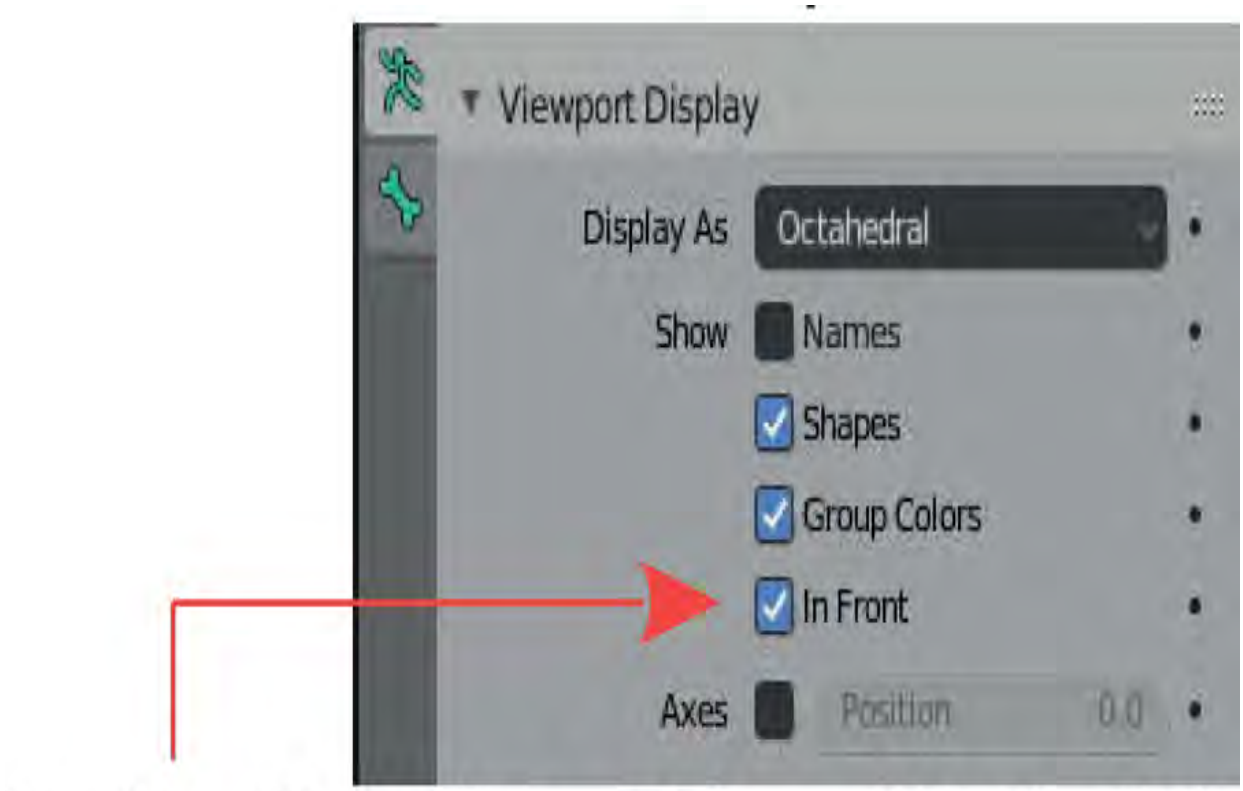


Figure 22.77



In the **Properties Editor, Object Data Properties, Viewport Display Tab,** check **In Front** (Figure 22.78).

Figure 22.78

Deselect the Tip of the Armature Bone and place the 3D View Editor in **Front Orthographic View** (Figure 22.78). Have the 3D Viewport Editor in **Edit Mode.** Additional Bones are to be added and you want them to be part of a Rig, that is, Parented to the Root Bone. The Root Bone and all additional Bones will be Parented forming a Rig assembly.

Adding More Bone

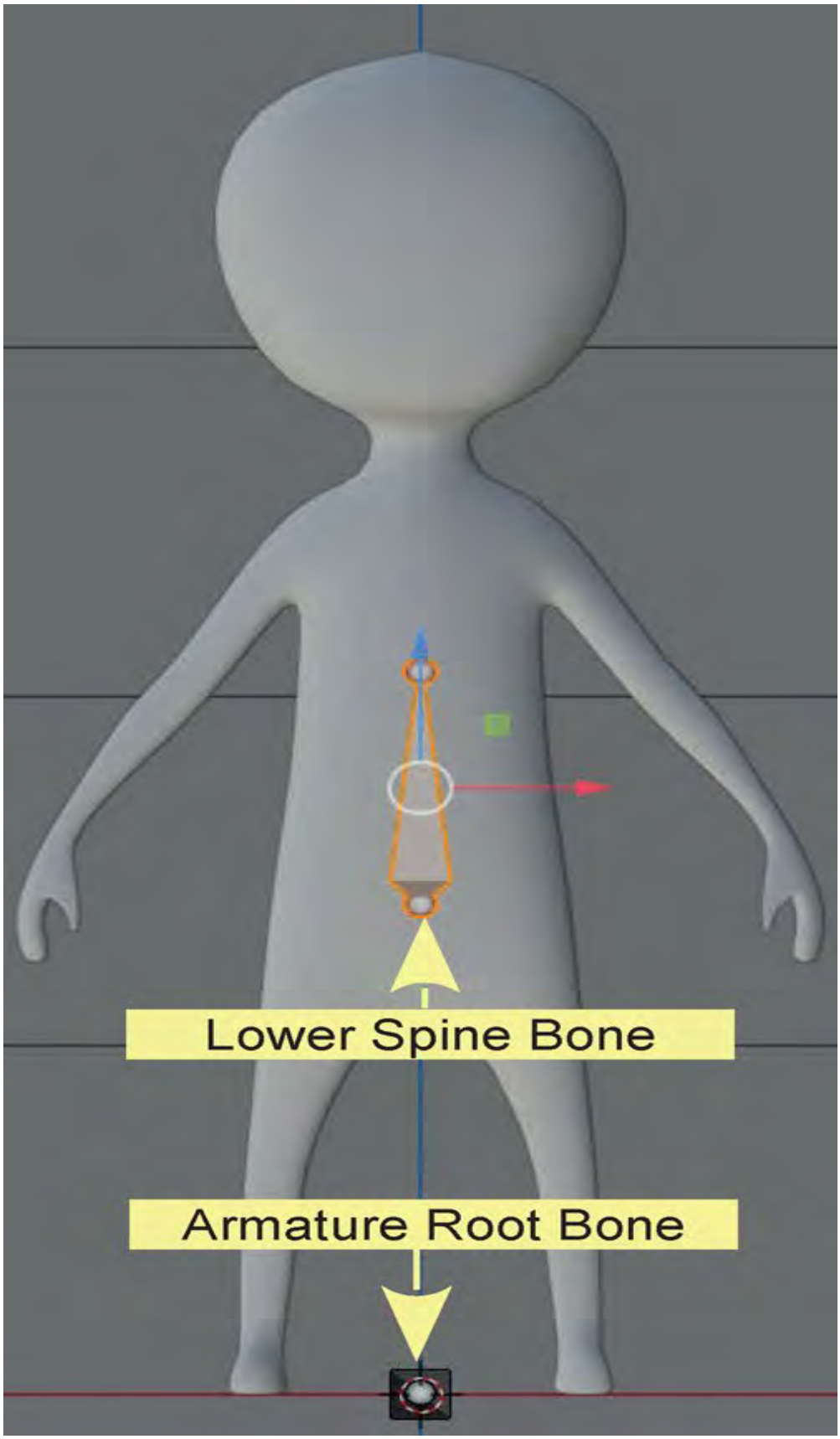

Lower Spine Bone
Armature Root Bone

Figure 22.79

A second Bone will be added to the Rig. Select the **Body** of the **Armature_Root Bone** in **Edit mode** then press **Shift + D key (Duplicate).** Drag the mouse and move the duplicated bone up to the pelvic area of the figure, rotate scale and position as shown in Figure 22.79 by selecting the Body of the Bone, or by selecting the Tip or the Base (in Edit Mode).

Remember: In the Properties Editor, Object Data buttons, Viewport Display tab (with the Bone selected) have **In Front** checked (ticked) so you can see the Bone inside the model.

The Base should be positioned approximately where the pelvis would be. Switch between Front Orthographic and Right Orthographic views to orientate the Bone. Consider this Bone to be the **Lower Spine Bone.**

The Armature_Root bone was **Duplicated in Edit Mode** to produce the Lower Spine because this second Bone has to be linked to the Armature_Root Bone and to be part of the Rig. **Duplicating in Object Mode would cause the new Bone to be an independent Armature not connected to the Rig.**

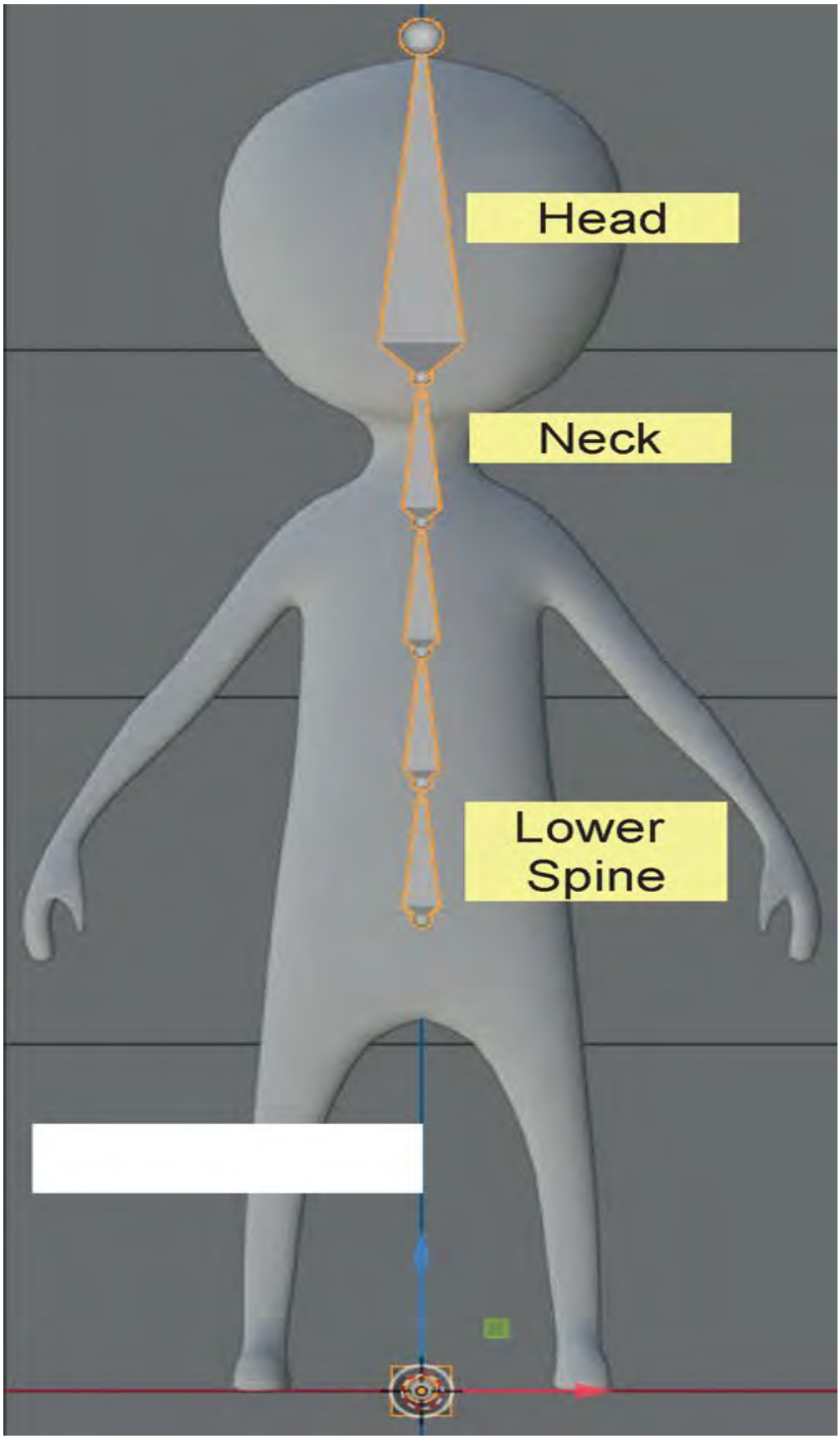

Head
Neck
Lower
Spine

Figure 22.80

When selecting either Armature_Root or Lower Spine in Object Mode both Bones will be selected. In Object Mode you are selecting the entire Rig not individual Bones. To select individually you have to be in Edit Mode.

With the Lower Spine Bone **in Edit Mode** move the Tip down to where the belly button would be then extrude Bones to form the remainder of spine (Figure 22.80). With the Tip of Lower Spine selected, press E key then Z and drag the mouse to extrude a new Bone (E key – extrude, Z key confines the extrusion to the Z axis.).

Repeat the process for each new Bone in the spine. Right Orthographic view allows you to position Bones to shape the spine. In Front Orthographic view the Bones follow the centerline of the figure. For the Head Bone restrain the extrusion to the Z axis by pressing E key + Z key.

Note: In positioning the Bones in Right Orthographic View it is not intended to replicate a human spine. Bones are placed to associate with parts of the Mesh model i.e. the Neck Bone will be linked to the Neck mesh. With Chibi, vertical Bones are fine.

Obviously there are many more Bones in a human spine than shown in the diagrams. In creating an Armature for animation it is good practice to minimise the number of Bones since this saves computer power in the animation process and simplifies the naming. The more Bones you have in a **Rig** the more flexible posing will be, therefore, you have to compromise.

Creating Arm Bone

Remember: Blender has the ability to mirror Bones to the opposite side of the model (Reference Section 22.8 Figure 22.19).

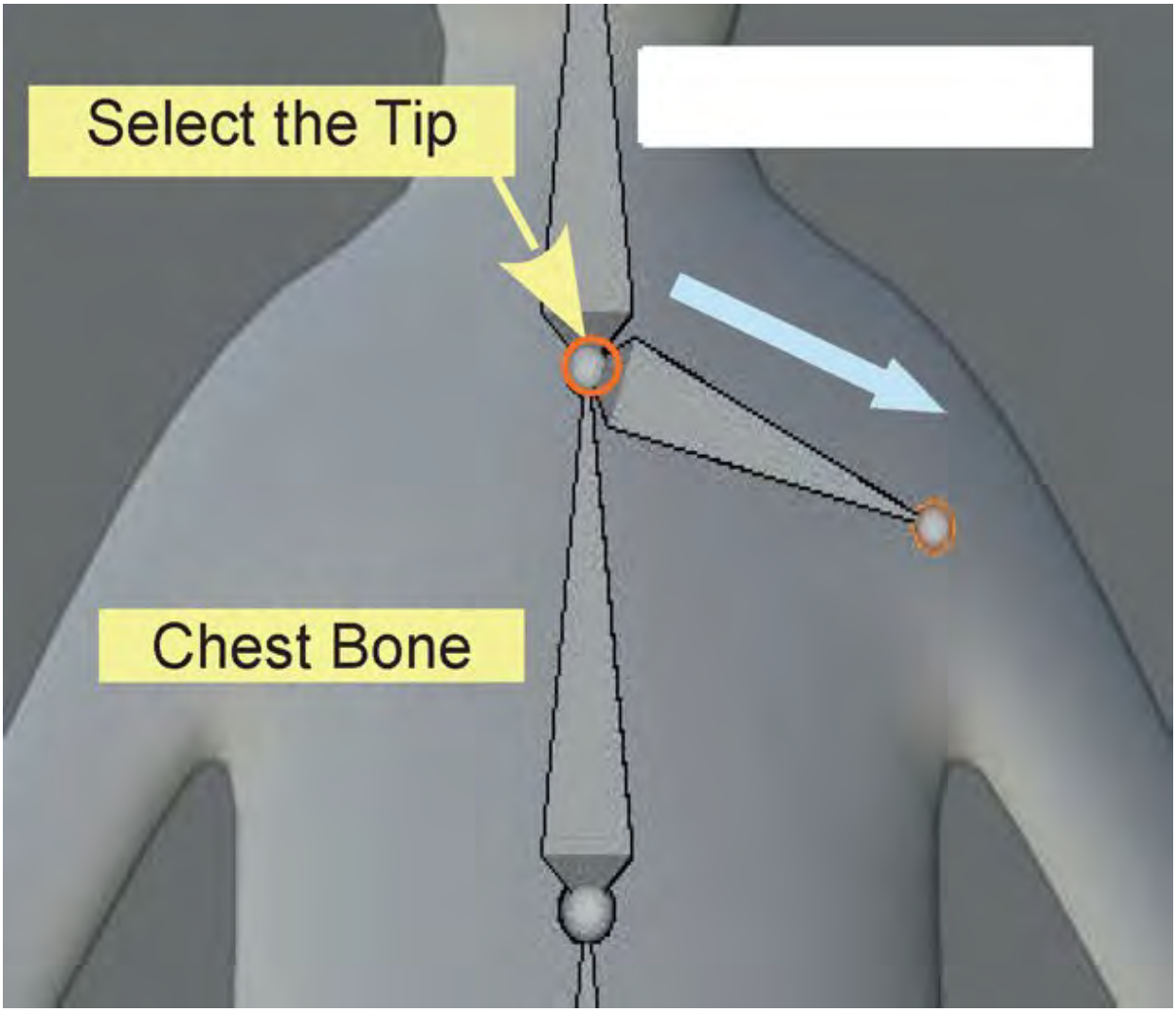


Figure 22.81

In **Edit Mode** select the Tip of the Chest Bone **in Front Orthographic view** and extrude a Shoulder Bone (Figure 22.81). Press the **N Key** for the **Properties Panel,** select **Tool** and check **X-Axis Mirror** in the panel.

Note: There is no immediate effect in the 3D Viewport Editor.

When extruding a Bone the **Extrude Panel** displays in the lower RHS of the Screen. With X-Axis Mirror checked in Armature Properties, check **Forked**

in the Extrude Panel (Figure 22.82) to duplicate the Bone on the opposite side of the Armature (Figure 22.83).

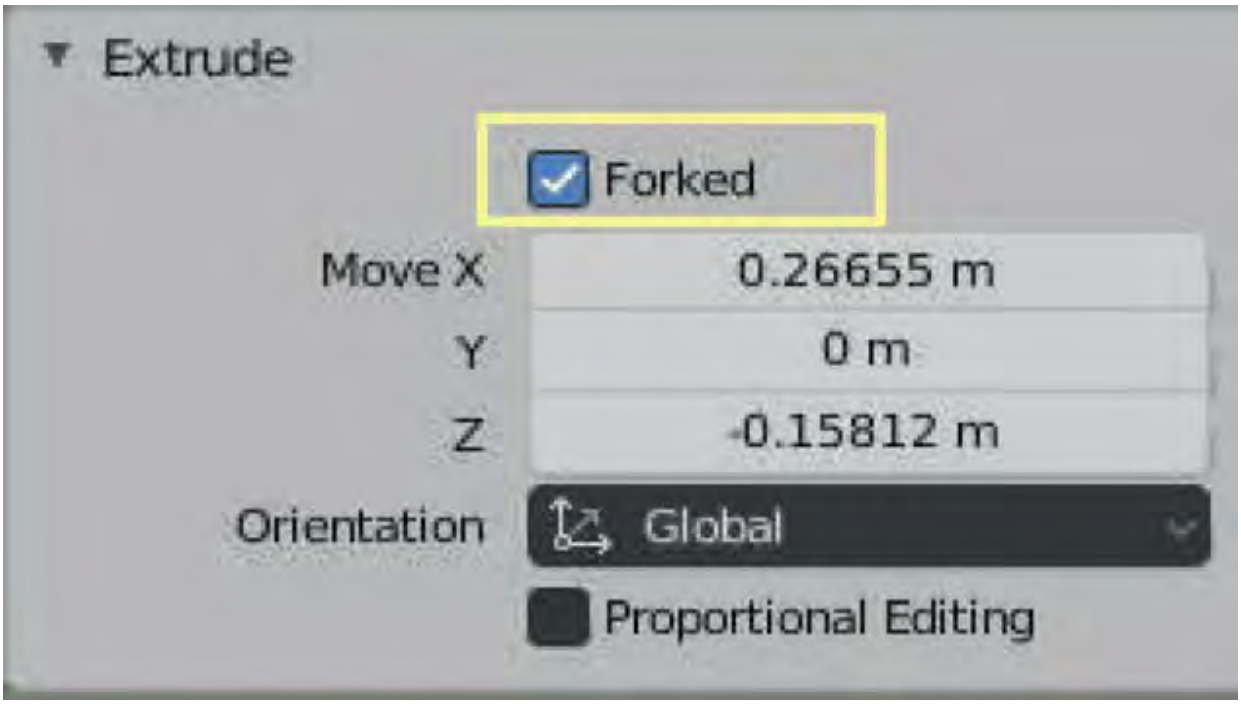


Figure 22.82

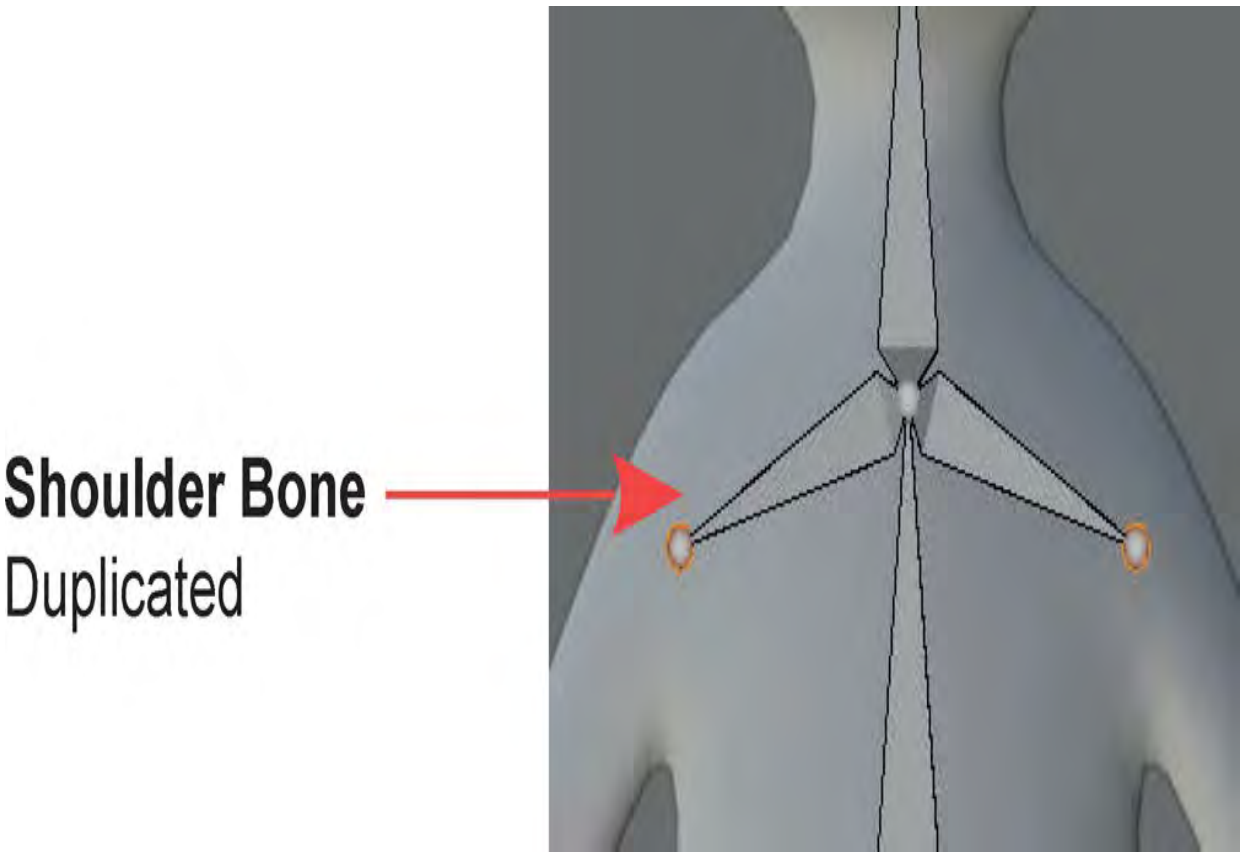


Figure 22.83

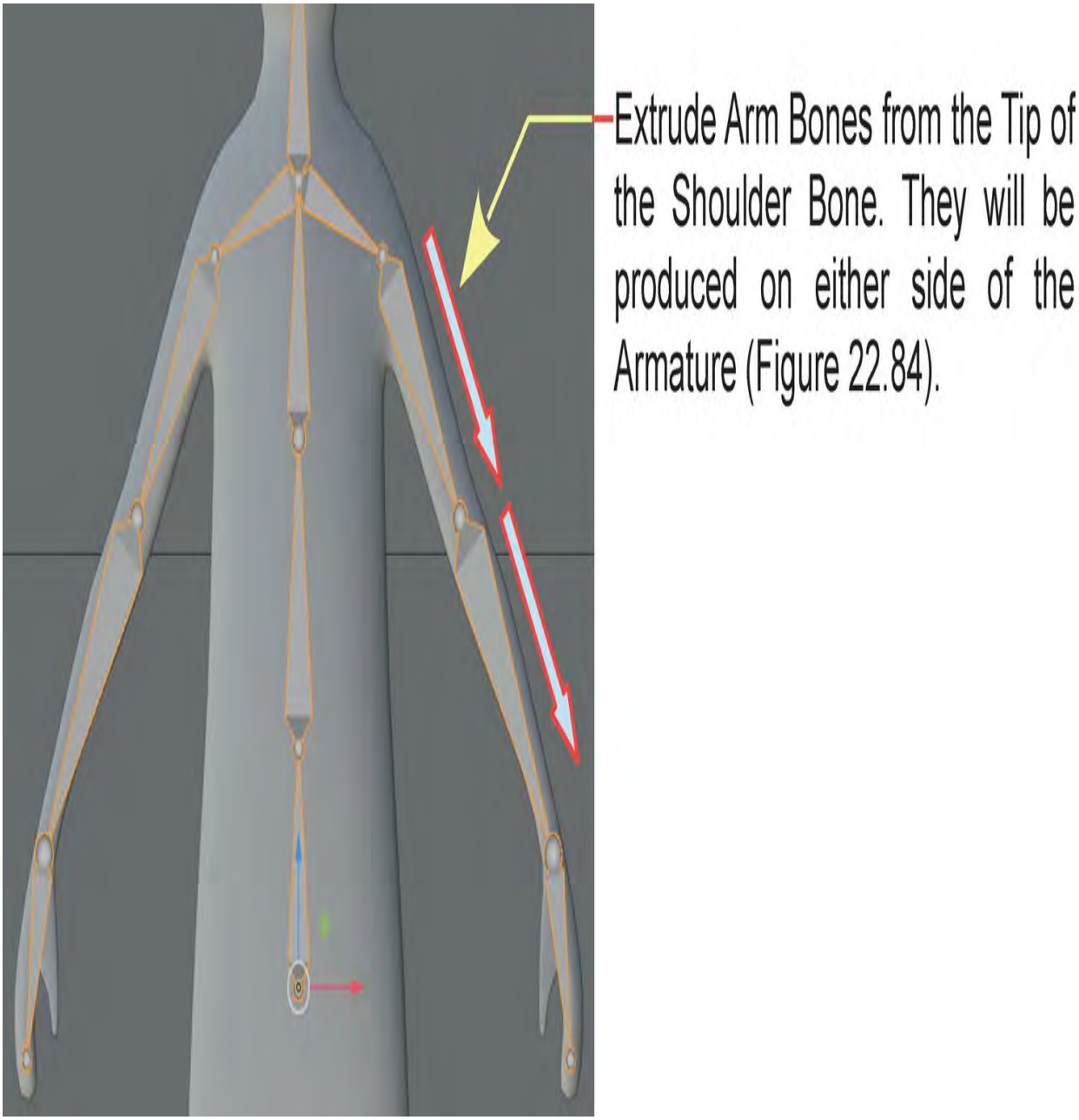

Extrude Arm Bones from the Tip of the Shoulder Bone. They will be produced on either side of the Armature (Figure 22.84).

Figure 22 84

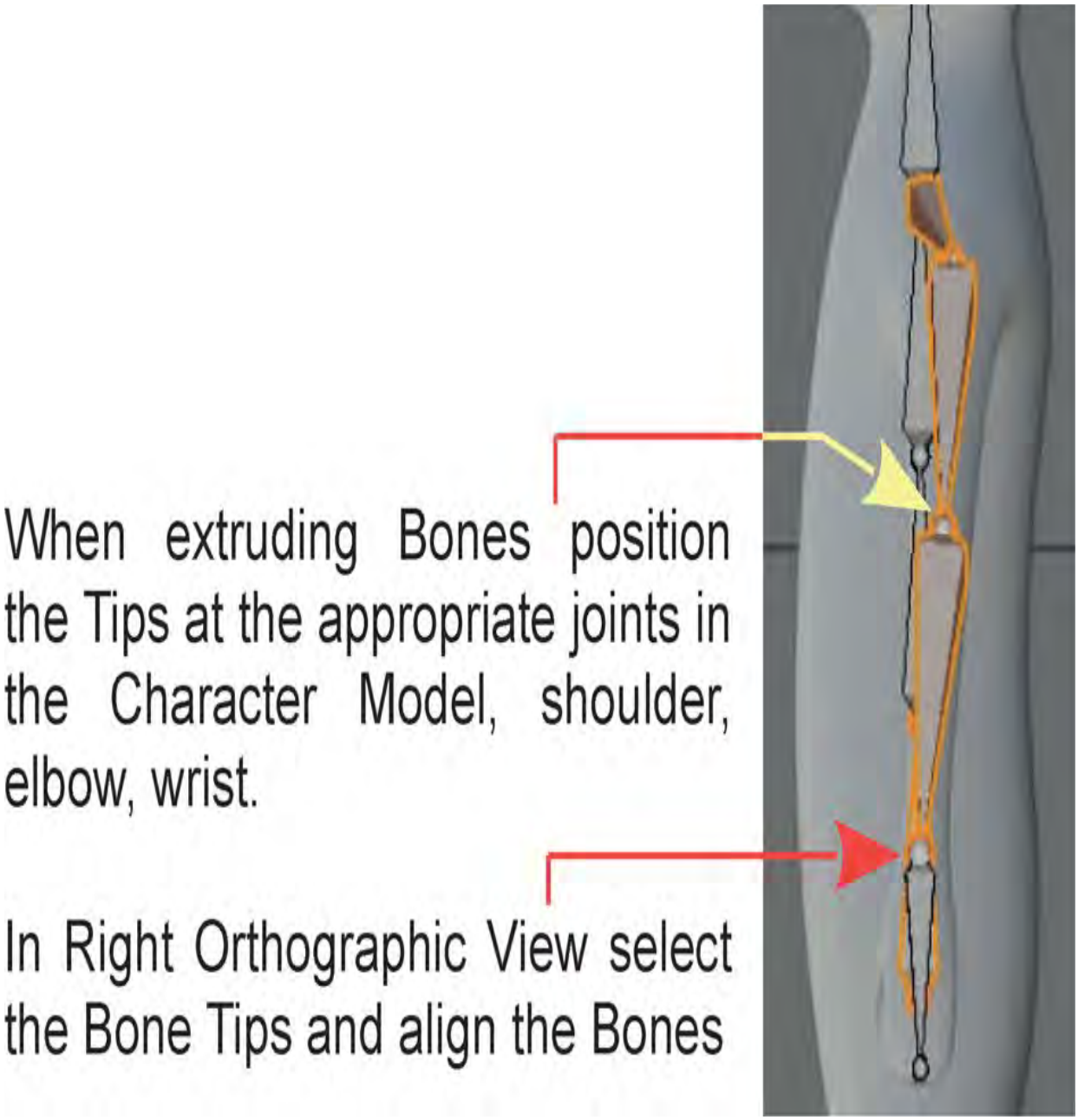
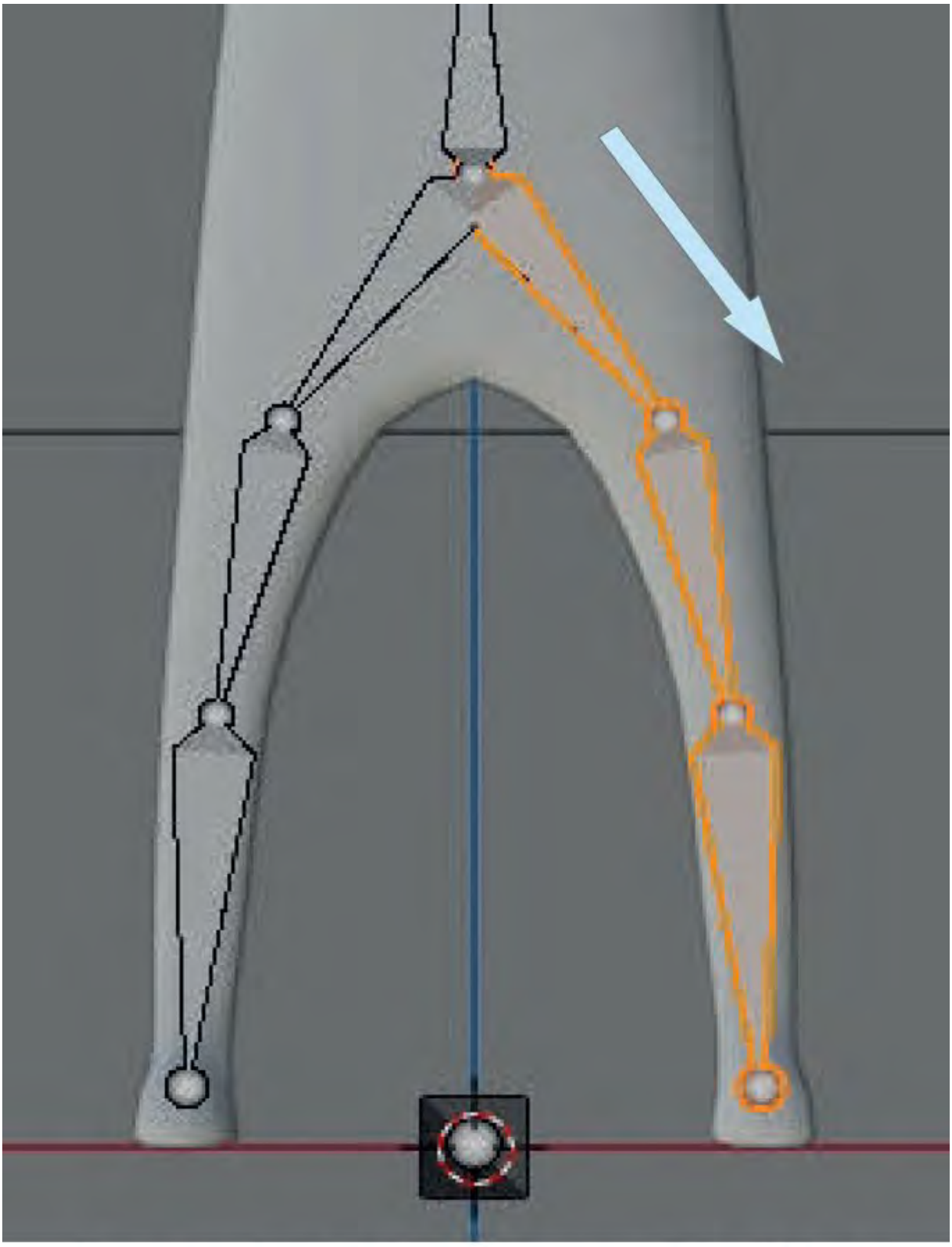


Figure 22.85

Creating Leg Bone

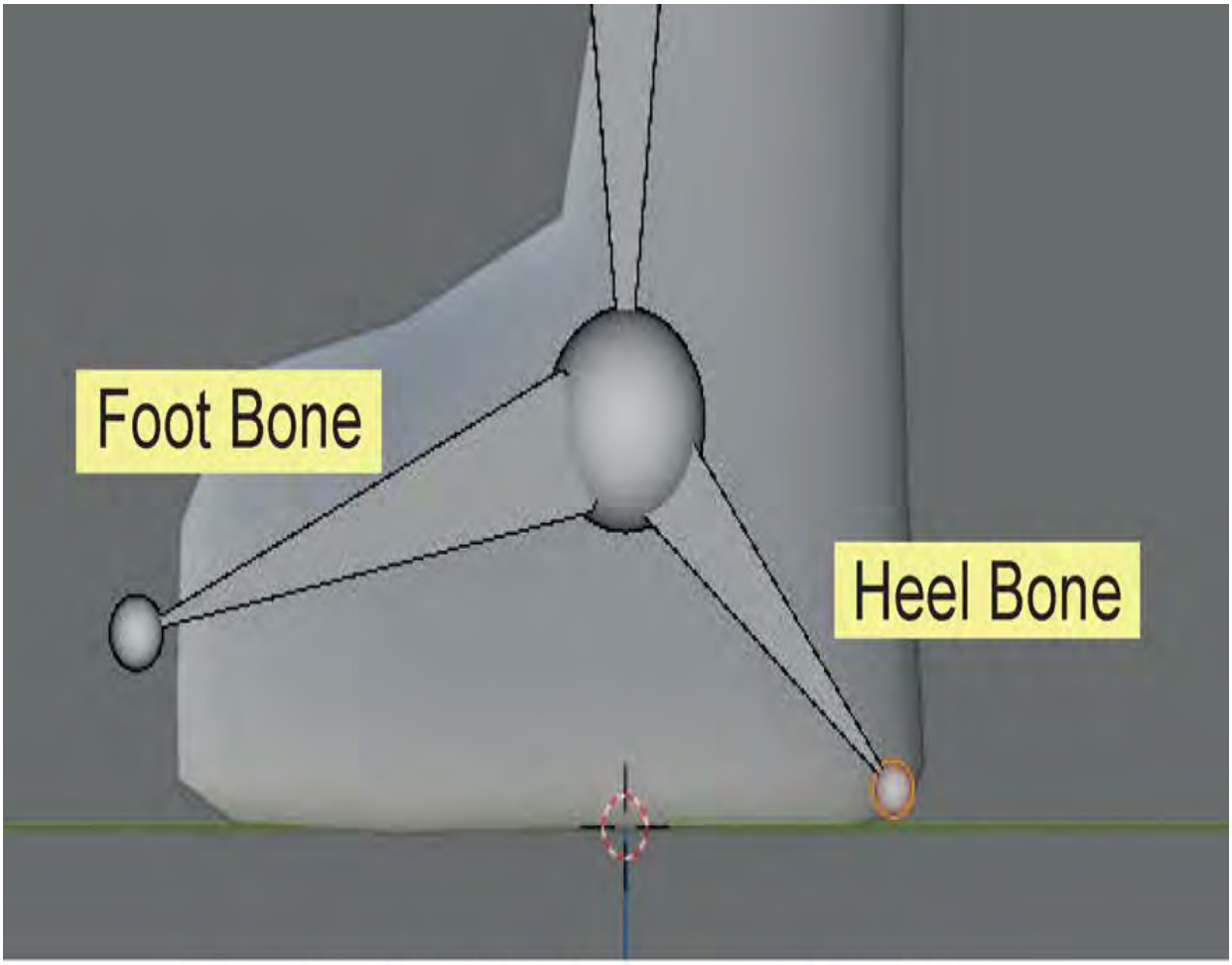
To add leg Bones for the figure repeat the process used for the arms, this time extruding from the Base of the Lower Spine Bone (Figure 22.86). **At the first extrusion check Forked in the Extrude Panel.**



Legs - Front Orthographic View

Figure 22.86

When you come to the Ankle go to Right Orthographic View and extrude a Foot Bone and a Heel Bone 22.87).



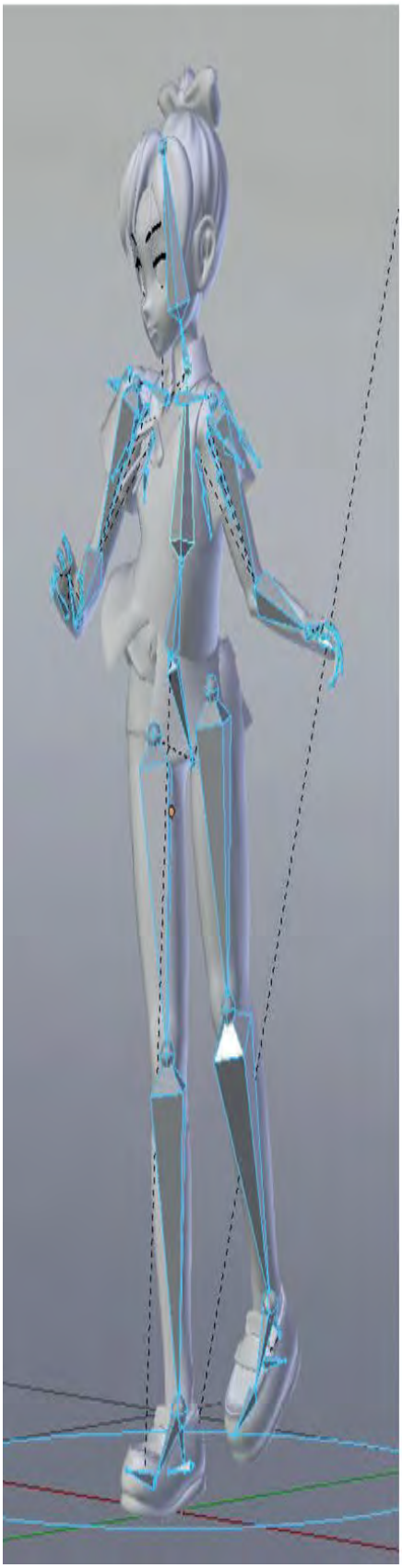
Foot – Right Orthographic View

Figure 22.87

Note: When positioning Bones, it is not intended to replicate a human skeleton. The individual Bones should be placed such that the Mesh Surface of the Model to be affected by a Bone is encapsulated in the Bone's Deform Envelope (Reference 22.11 Parenting with the Armature).

The Rig so Far

By comparing the Armature Rig so far with the introductory image (Figure 22.1) you will see where you are heading.



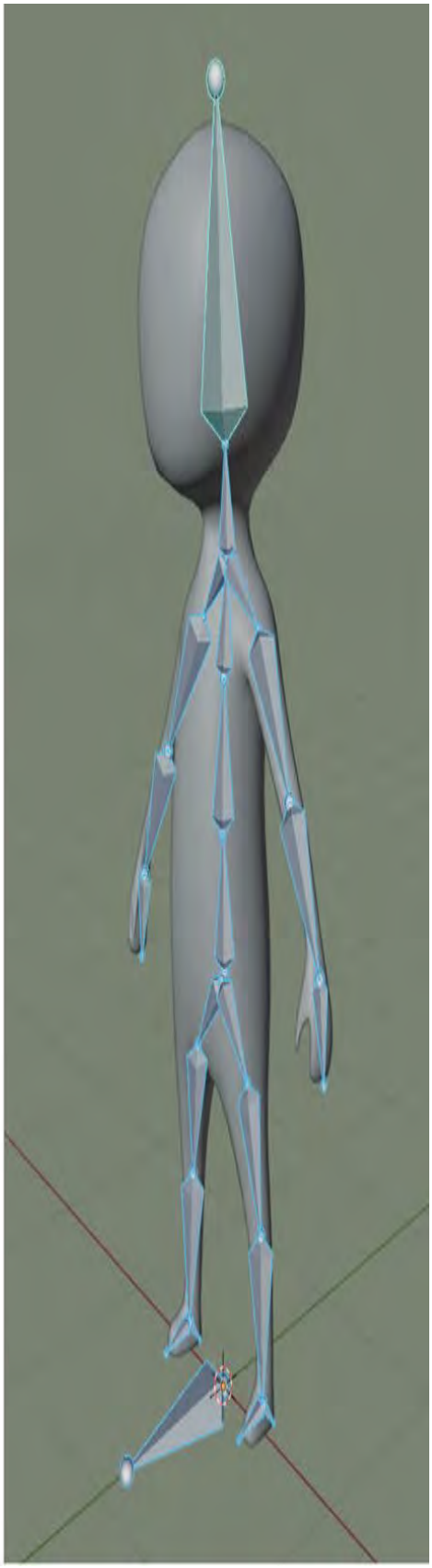


Figure 22.88

22.19 Bone Naming

In this relatively simple Armature. The individual Bones have been referred to as, Armature Root, Lower Spine, Neck, Head etc. but there is nothing in the View to specifically indicate which Bone is which. Naming Bones and giving the Bones meaningful names is very important, especially in a complicated Rig.

Blender automatically names Bones as they are extruded but the naming system provides names such as Bone, Bone.001, Bone.002 etc. and when Mirroring is involved, Bone.002_R and Bone.002_L. You see the names in the Outliner Editor.

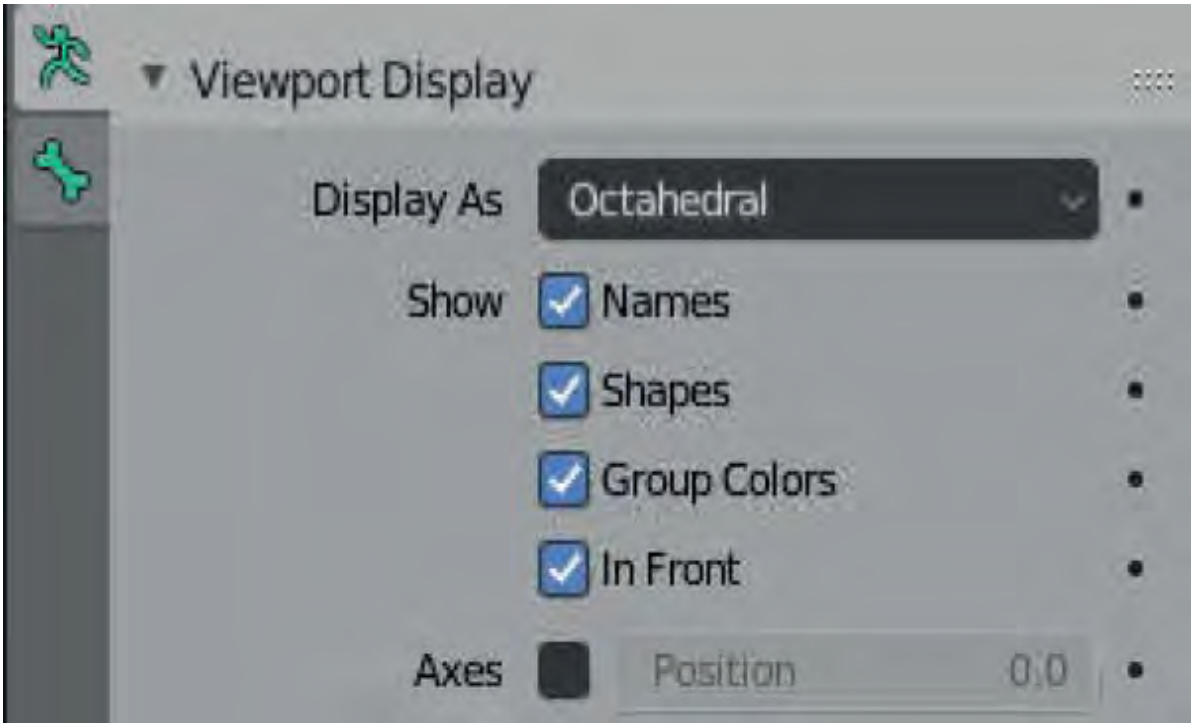


Figure 22.89

Bone Names may be displayed in the 3D Viewport Editor by checking **Names** in the **Properties Editor, Object Data Properties, Viewport Display**

Tab for the Armature (Figure 22.89). The default names are displayed adjacent to each Bone in the Armature in the **3D Viewport Editor** (Figure 22.90) and a hierarchy listing is shown in the **Outliner Editor** when in Edit Mode or Object Mode (Figures 22.91). **Note: The listing does NOT display in Pose Mode.**

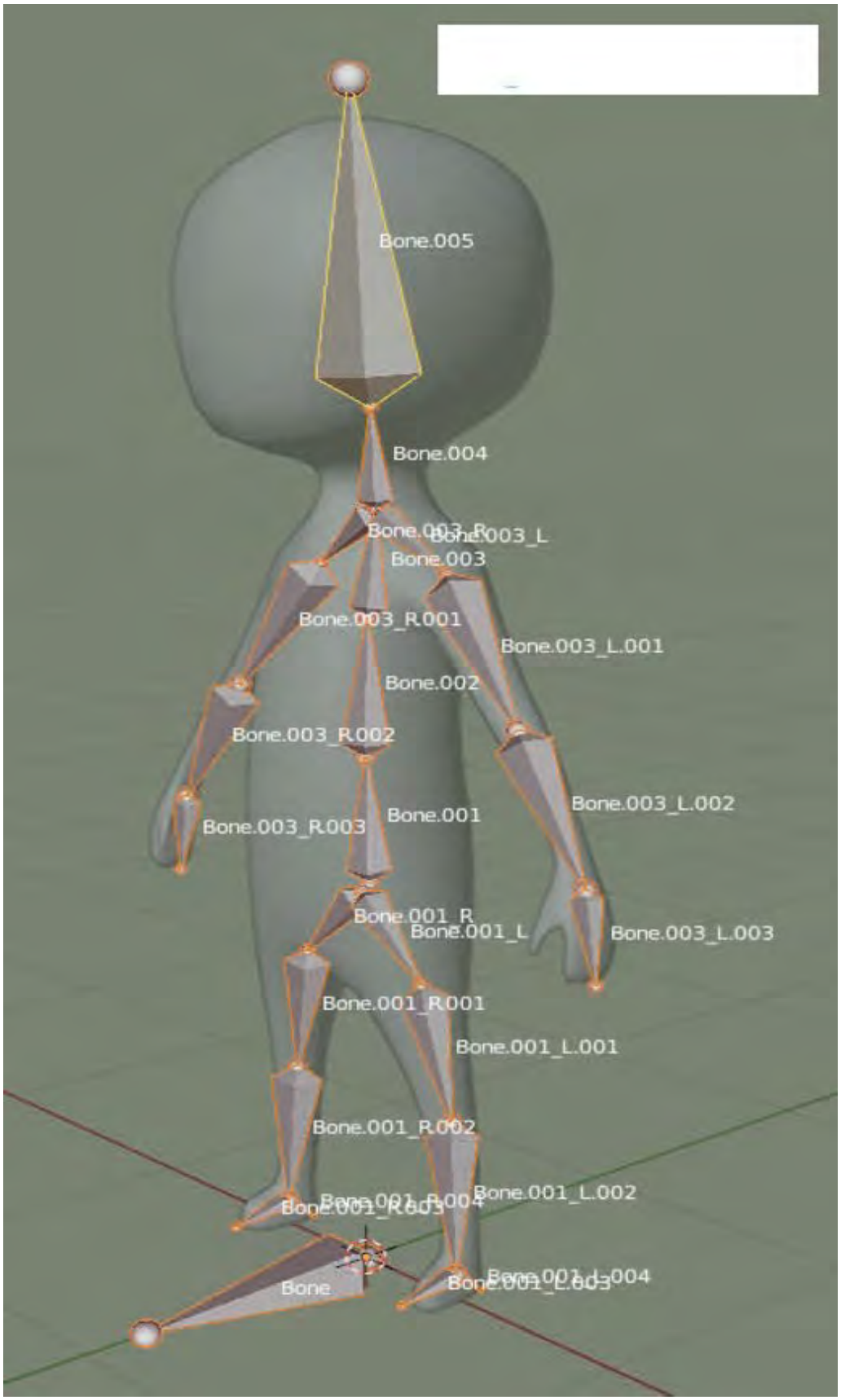

Bone.005
Bone.004
Bone.003_R
Bone.003_L
Bone.003
Bone.003_R.001
Bone.003_L.001
Bone.002
Bone.003_R.002
Bone.003_L.002
Bone.003_R.003
Bone.001
Bone.001_R
Bone.001_L
Bone.003_L.003
Bone.001_R.001
Bone.001_L.001
Bone.001_R.002
Bone.001_L.002
Bone.001_R.003
Bone.001_R.004
Bone
Bone.001_L.003
Bone.001_L.004

Figure 22.90

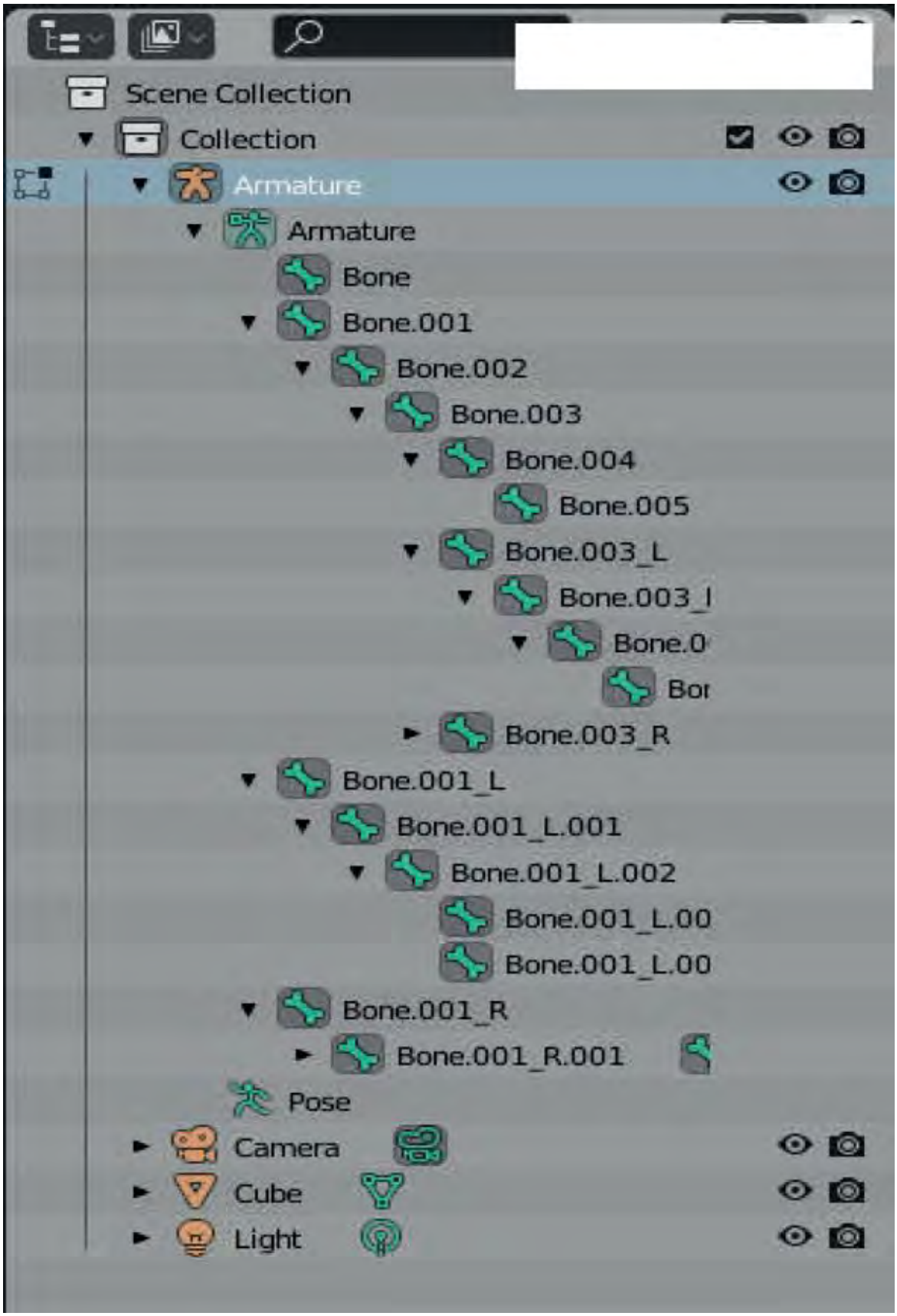

Scene Collection
Collection
Armature
Armature
Bone
Bone.001
Bone.002
Bone.003
Bone.004
Bone.005
Bone.003_L
Bone.003_R
Bone.001_L
Bone.001_L.001
Bone.001_L.002
Bone.001_R
Bone.001_R.001
Pose
Camera
Cube
Light

Figure 22.91

Individual Bones can be renamed by selecting a Bone in Edit Mode then editing the Bone name in the Properties Editor, Bone Properties .

Alternatively click on the name in the Outliner Editor (Figure 22.91) to select the Bone. Double click to edit the name.

22.20 Assigning the Mesh

At this point, although the **Armature Rig** is incomplete, you may assign it to the **Mesh Figure.**

This is the process of linking Vertex Groups (groups of vertices) on the figure's mesh surface to individual Bones. **Blender has an automated process for doing this.** The Bones will then control the posing (posturing) of the mesh.

Figures 22.90 and 22.91 show the Bone names in the 3D Viewport and in the Outliner Editors.

In Figure 22.87 the 3D Viewport Editor is shown in Edit Mode. To see the full list of all the Bones in the Outliner Editor expand each entry.

Note: Bones are named according to Chibi's Left and Right not your's.

Before engaging the automated mesh assigning process you need to exclude the **Root_Bone.** This bone is a **Control Bone** for moving the figure around in the Scene and is not a **Posing Bone.** Posing is the process of posturing the figure. In Figures 22.90 and 22.91 the Root Bone is named Bone. **Rename it to Root_Bone.**

In the 3D Viewport Editor in **Edit Mode** select **Root_Bone.** In the **Properties Editor, Bone Properties, Deform tab** click on the **Deform**

button to remove the tick. This tells Blender that you do not want **Root_Bone** to be part of the deforming Rig.

In the **3D Viewport Editor** deselect the bone and change to **Object Mode.** Deselect the Armature Rig.

Assigning Mesh to Armature

Select Chibi (the Model) in Object Mode and if you haven't done so, in the Properties Editor, Modifiers buttons, click **Apply** in the Mirror and Subsef Modifiers to permanently assign the Modifiers. Both Modifiers will disappear from the Properties Editor.

Important: Before assigning the Mesh to the Armature make sure the individual Bones are positioned inside the parts of the Mesh. Check the Front and Right Orthographic Views.

With the **Mesh figure (Chibi)** selected **Shift** select the **Armature Rig (** LMB click the Figure, hold Shift, LMB click on the Armature protruding from the Head). With the mouse cursor in the 3D Viewport Editor press **Ctrl + P key** to display the **Set Parent To** menu and select the **With Automatic Weights** option.

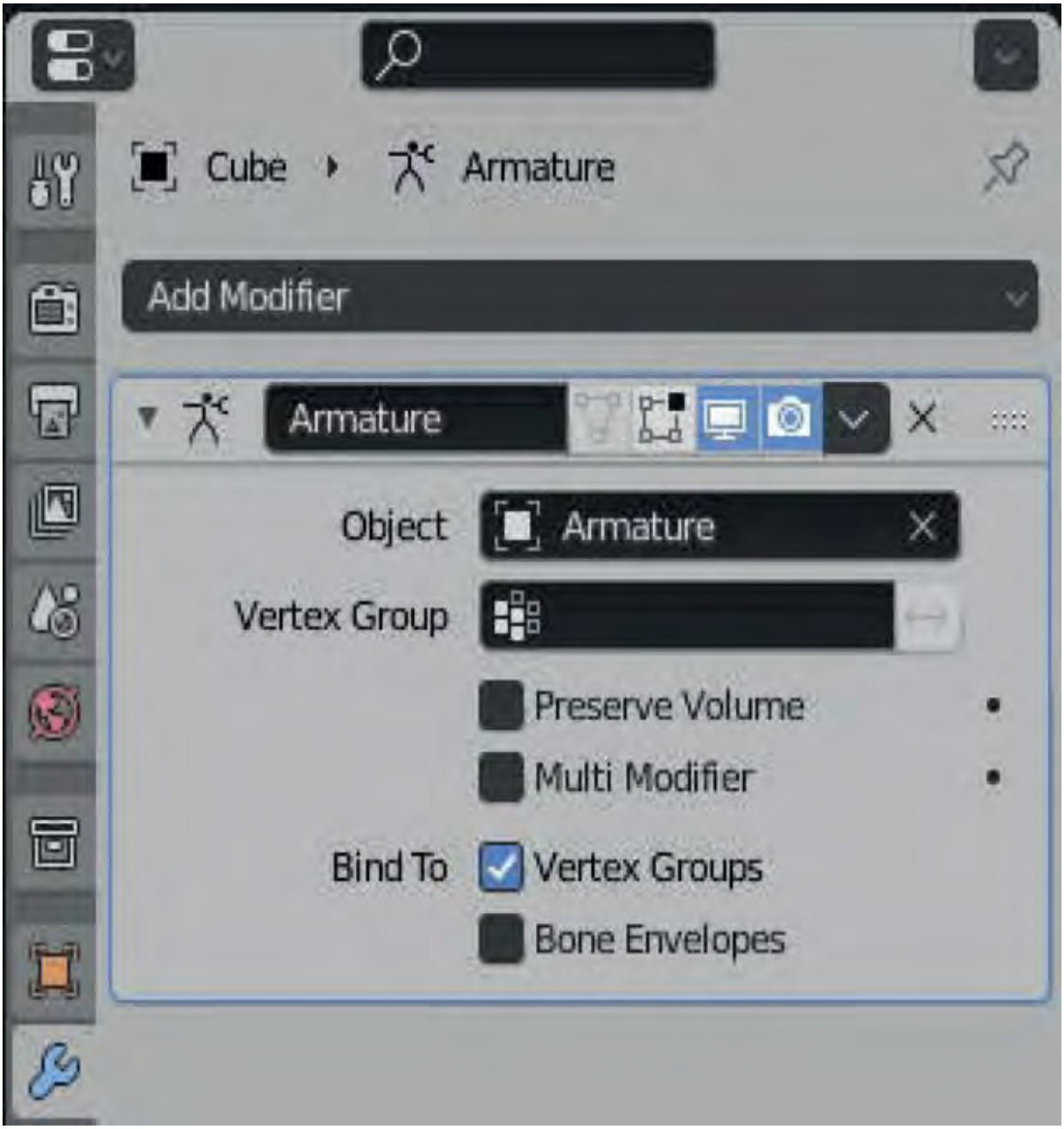


Figure 22.92

Note: In assigning Automatic Weights a new **Armature Modifier** is assigned to the Mesh Figure in the Properties Editor (Figure 22.92) (deselect and re select Chibi in Object Mode to see the Modifier).

Deselect the Mesh Figure and the Armature in the 3D Viewport Editor then select the Armature only. Go into **Pose mode** and select and rotate individual

Bones to pose the figure (Figure 22.93) - Bones **Lower_Arm_L** and **Upper_Arm_L** are rotated).

22.21 Vertex Groups

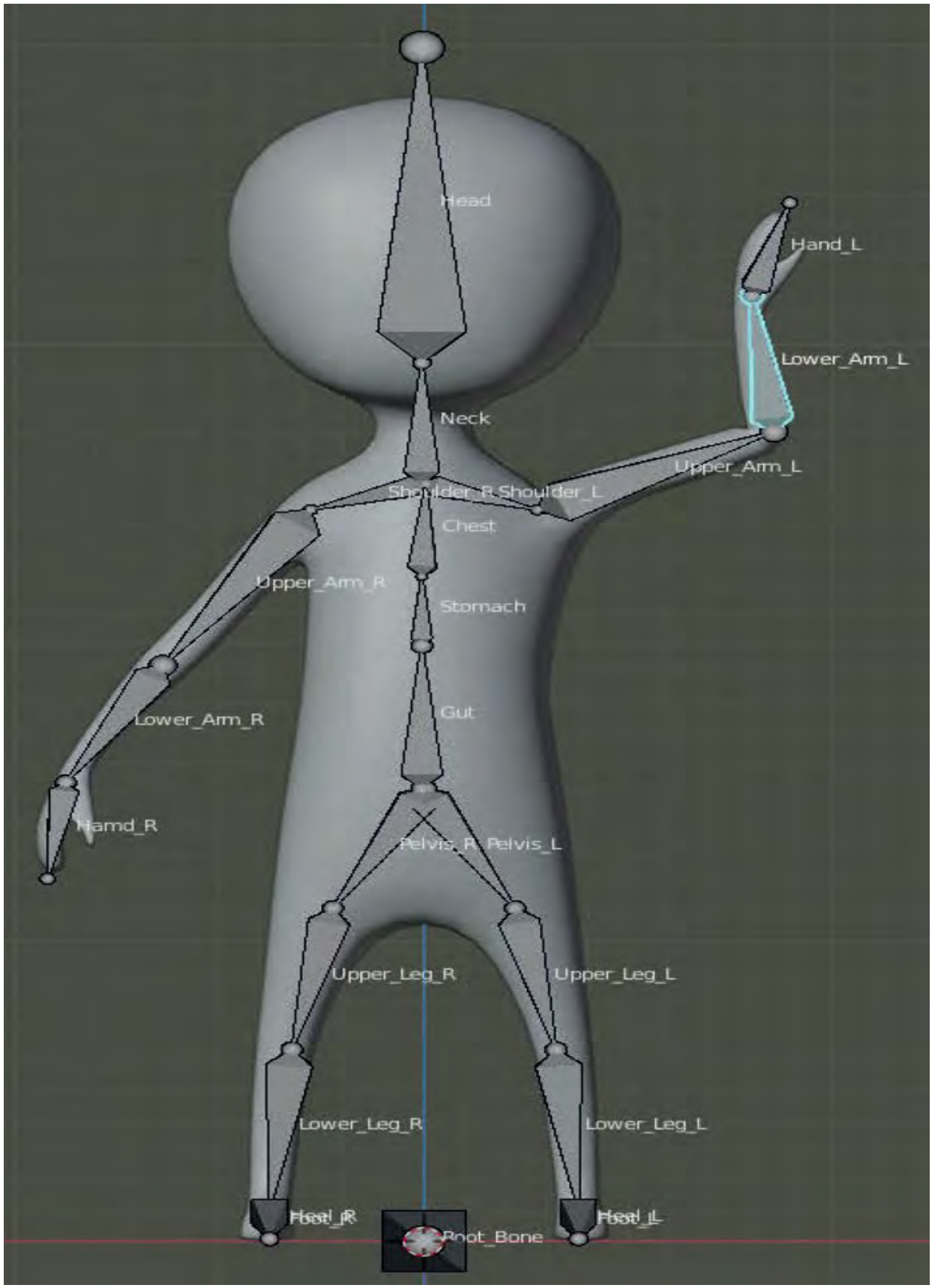

Head
Hand_L
Lower_Arm_L
Neck
Upper_Arm_L
Shoulder_R
Shoulder_L
Chest
Upper_Arm_R
Stomach
Lower_Arm_R
Gut
Hamd_R
Pelvis_R
Pelvis_L
Upper_Leg_R
Upper_Leg_L
Lower_Leg_R
Lower_Leg_L
Root_Bone

Figure 22.93

When you employ the automatic Mesh Assignment, two things occur:

An Armature Modifier is added to the Mesh (Figure 22.94) and Vertex Groups are created in the Mesh and assigned to each Bone in the Armature.

With the mesh selected in **Object Mode** go to the **Properties Editor, Object Data Properties, Vertex Groups Tab** and you will see the **Vertex Groups (**Figure 22.94) There is a scroll bar at the RHS of the Vertex Group panel or expand the panel).

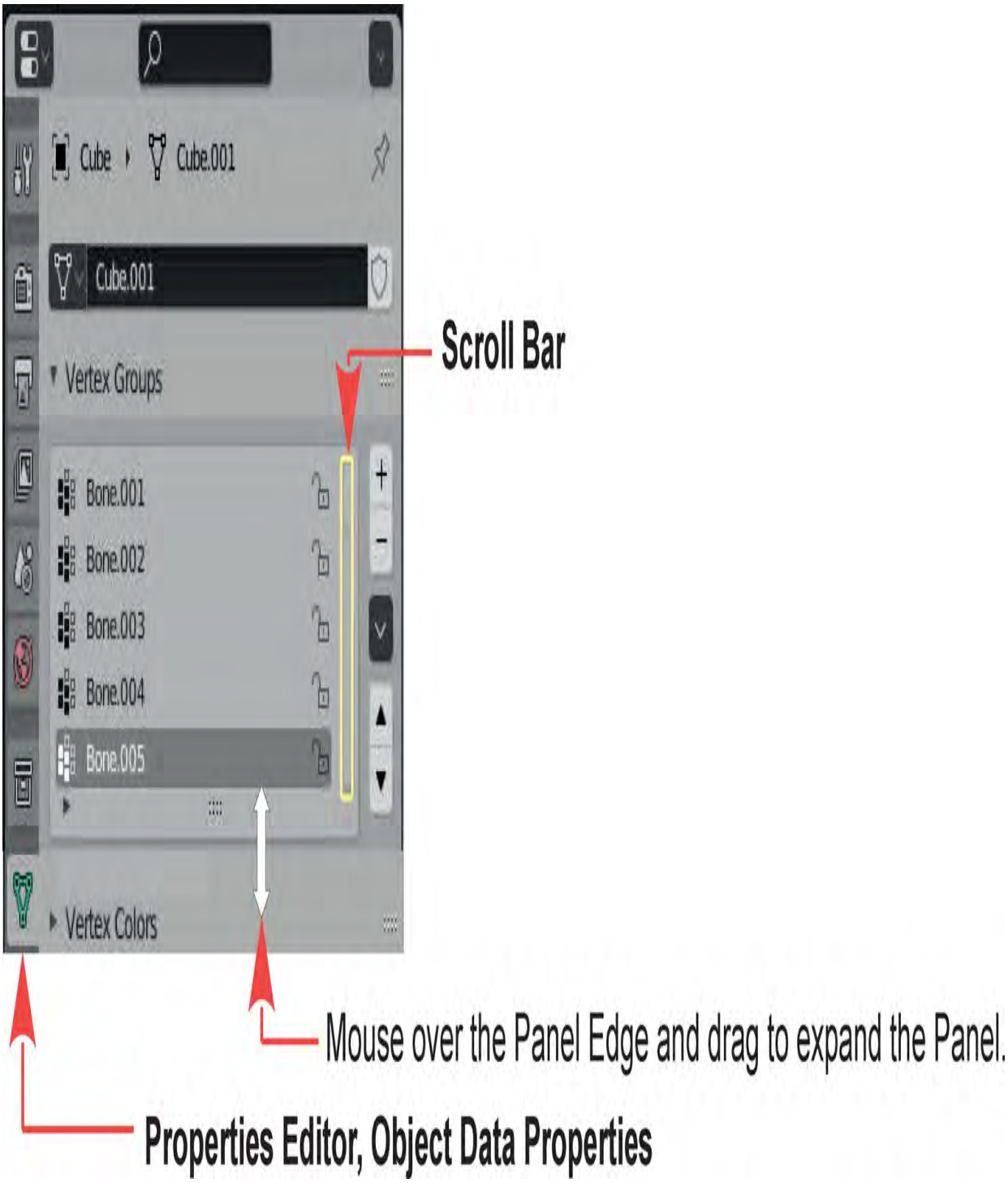


Figure 22.94

The Vertex Groups may be used for correcting incorrect Mesh Deformation. This is accomplished by selecting a **Vertex Group** and employing the **Weight**

Paint Tool to clean up the connections between the Mesh and the Vertex Groups (refer to the section on Weight Painting).

> **Note:** After posing in **Pose mode,** Bones will be returned to their original positions (Reset), individually by selecting each bone or collectively by selecting all Bones and pressing **Alt + R key** (Reset rotation) and **Alt + G key** (Reset Location - Resets Grab).

In a complicated Rig, as well as incorrect Mesh assignment, there may be Parenting issues which require addressing. Detailed tutorials provide instruction for correcting such issues but at this stage just be aware that automatic processes do not always provide perfect results.

22.22 Posing the Character Model

Posing a Model may be simply to give a character attitude when creating a still image but is mainly employed to create Keyframes in an animation sequence (see Chapter 20 - 20.6).

Even with a relatively simple Character Rig, posing individual Bones to create **Keyframes** can be a tedious process. There are semi automated procedures such as using an **Inverse Kinematics Constraint** (see 22.15) with a Hand or Foot such that when either is moved the Arms or Legs follow.

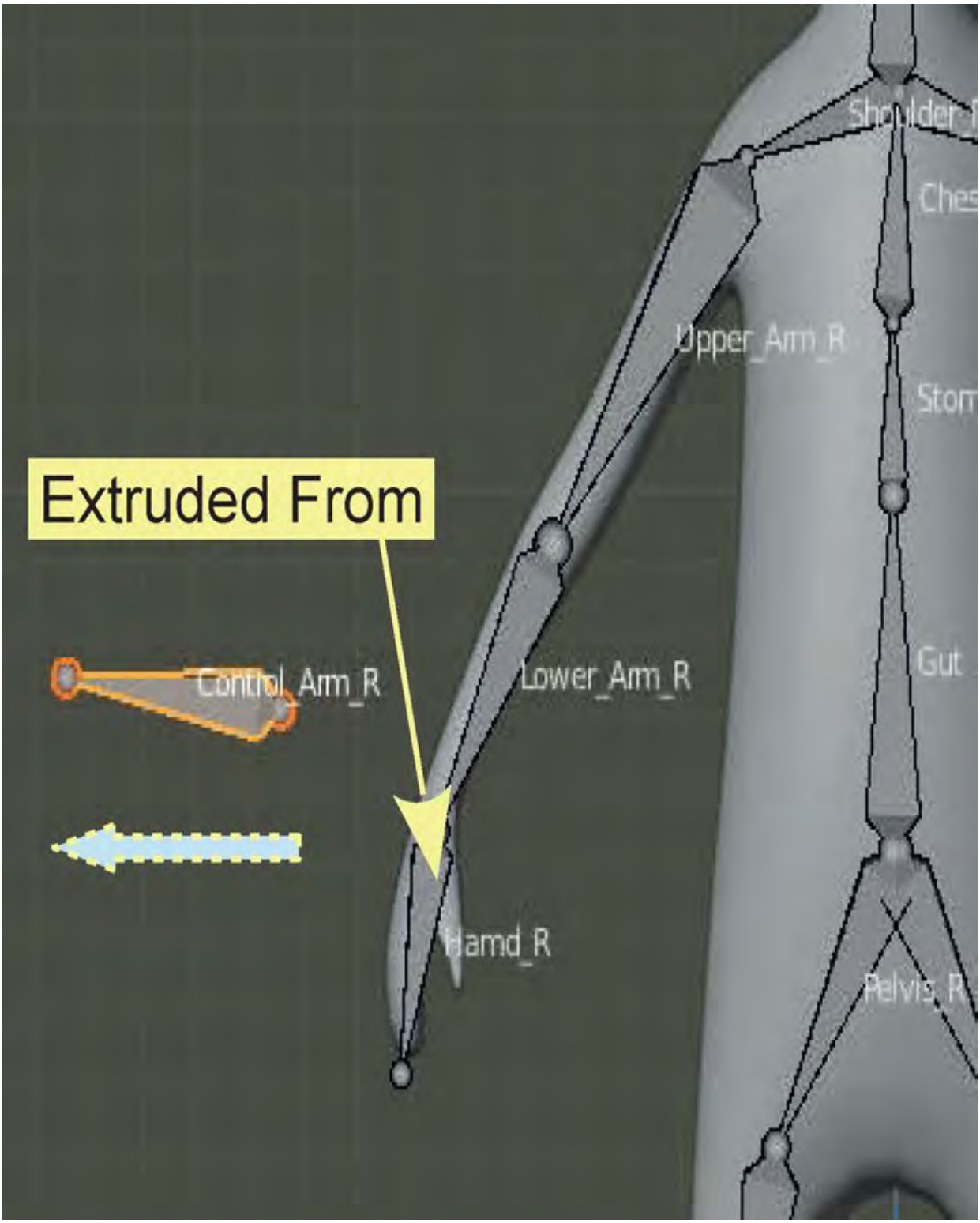


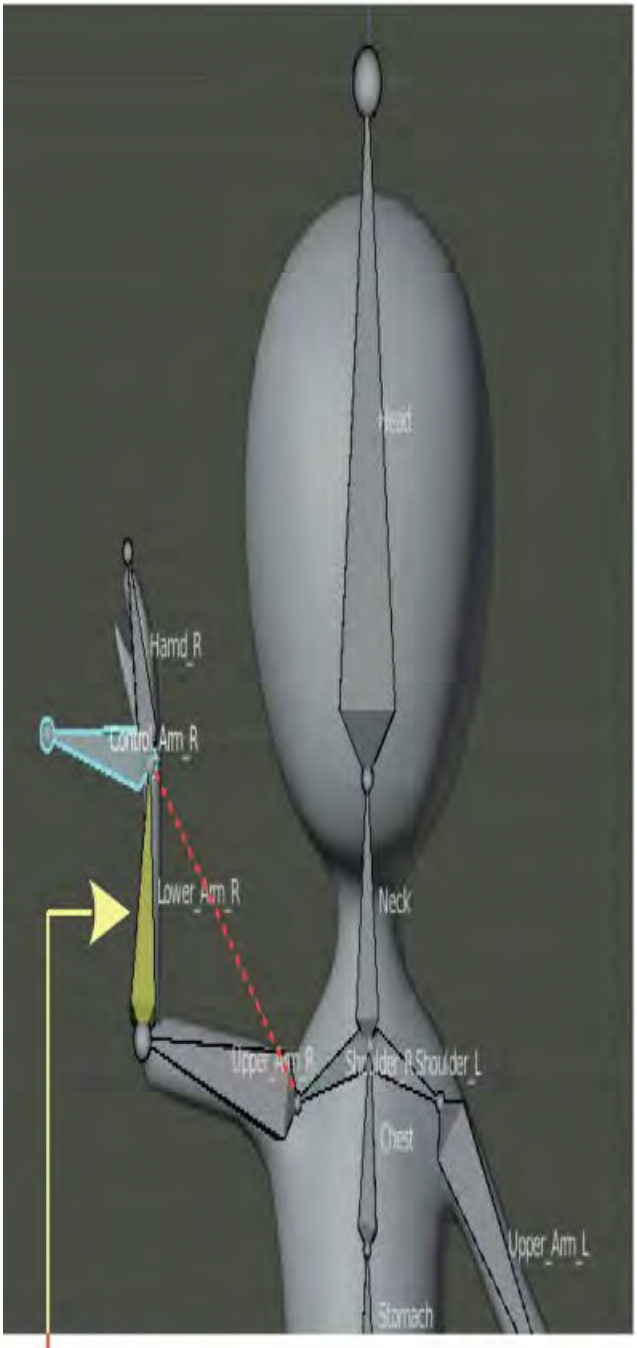
Figure 22.95

In Figure 22.95 a Control Bone has been extruded from the Tip of Lower_Arm_R. The Parenting has been cleared (Press ALT + P Key) and the Control Bone moved aside in Edit Mode. The Bone has been renamed **Control_Arm_R.**

Figures 22.96 shows an **Inverse Kinematic Constraint** applied to Lower_Arm_R **when in Pose Mode.** The Target in the Constraint is set as Armature with Bone set to Control_Arm_R. (See 20.15)

Control_Arm_R has been renamed in the Outliner Editor.

The **Chain Length** in the Constraint is **2** making the Constraint effective only to Lower_Arm_R and UpperArm_R.



Lowe_Arm_R displays olive green indicating that a constraint has been applied.

Bone Constraints Properties

Do not confuse Bone Constraint Properties with Object Constraint Properties.

Figure 22.96

Selecting the Control Bone, **Control_Arm_R** in Pose Mode and Translating allows the left arm to be Posed. Control Bones would be generated for the arm and the in the same manner and in fact for of the Armature you wish to Pose.

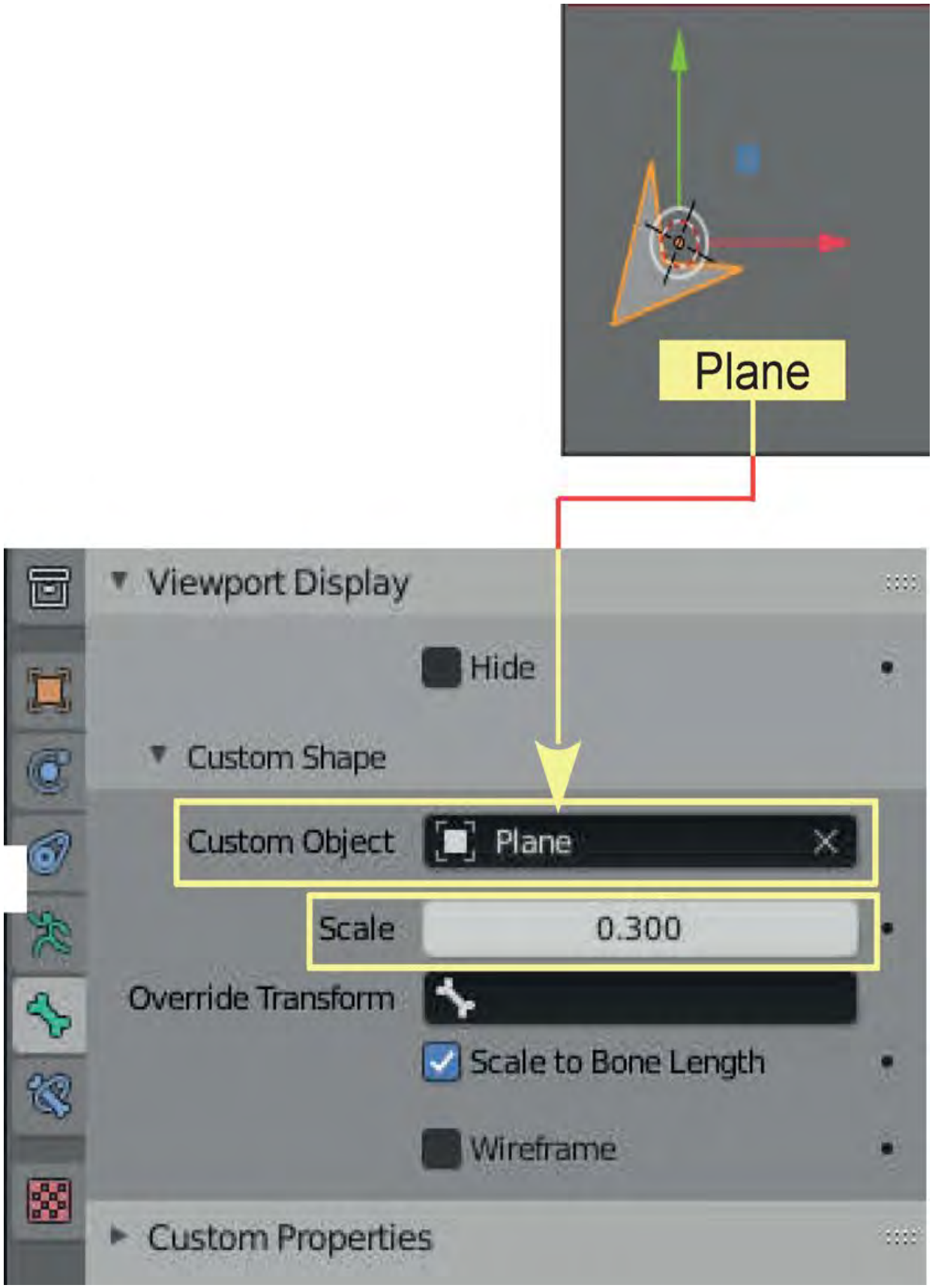

Plane
Viewport Display
Hide
Custom Shape
Custom Object
Plane
Scale
0.300
Override Transform
Scale to Bone Length
Wireframe
Custom Properties

Figure 22.97

Control Bones may be displayed as different individual shapes to distinguish from the Bones in the Armature. In the 3D Viewport Editor, create a shape from one of Blenders Primitives. Figure 22.97 shows a Plane Object reshaped. Park the Object to one side in the Scene.

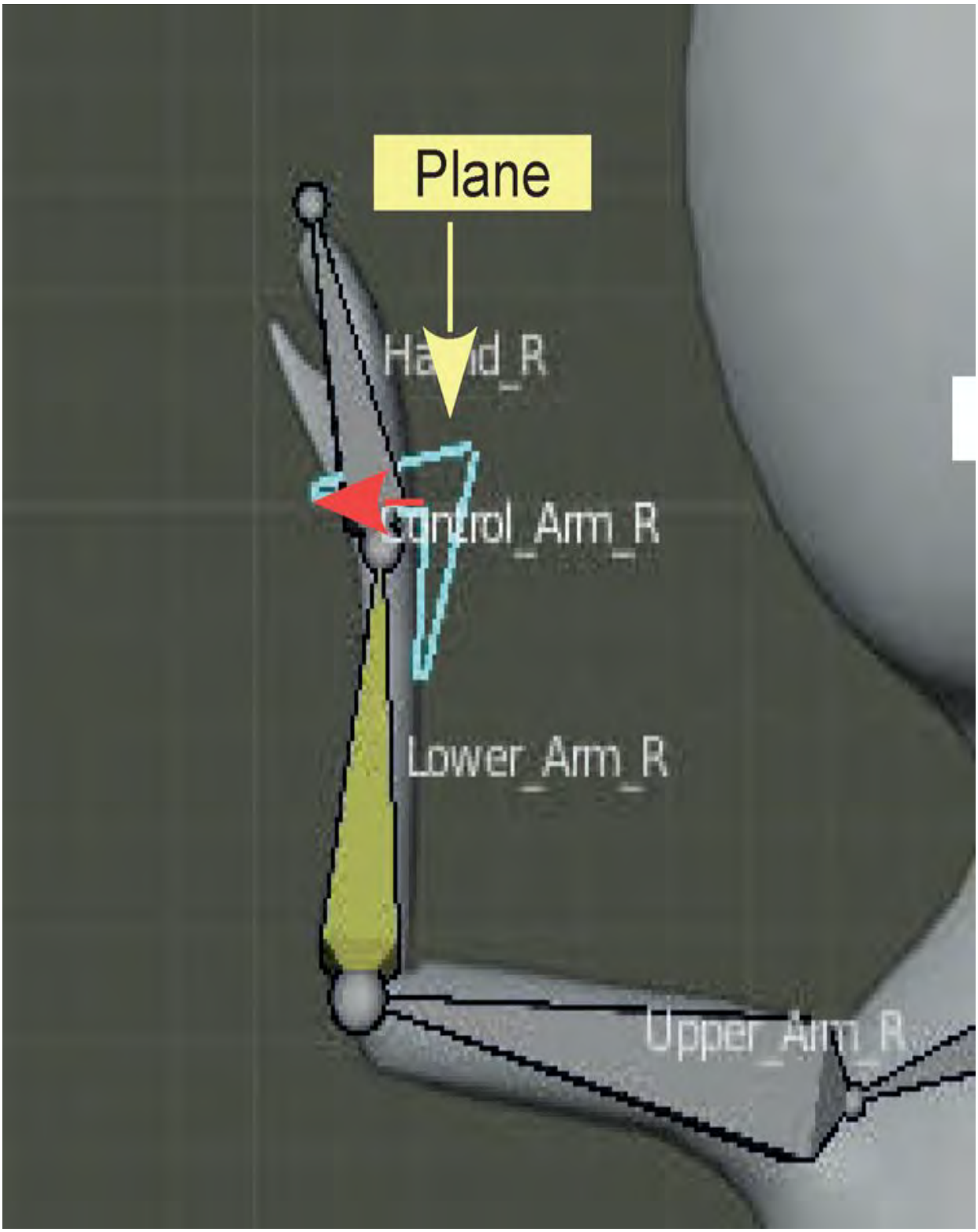
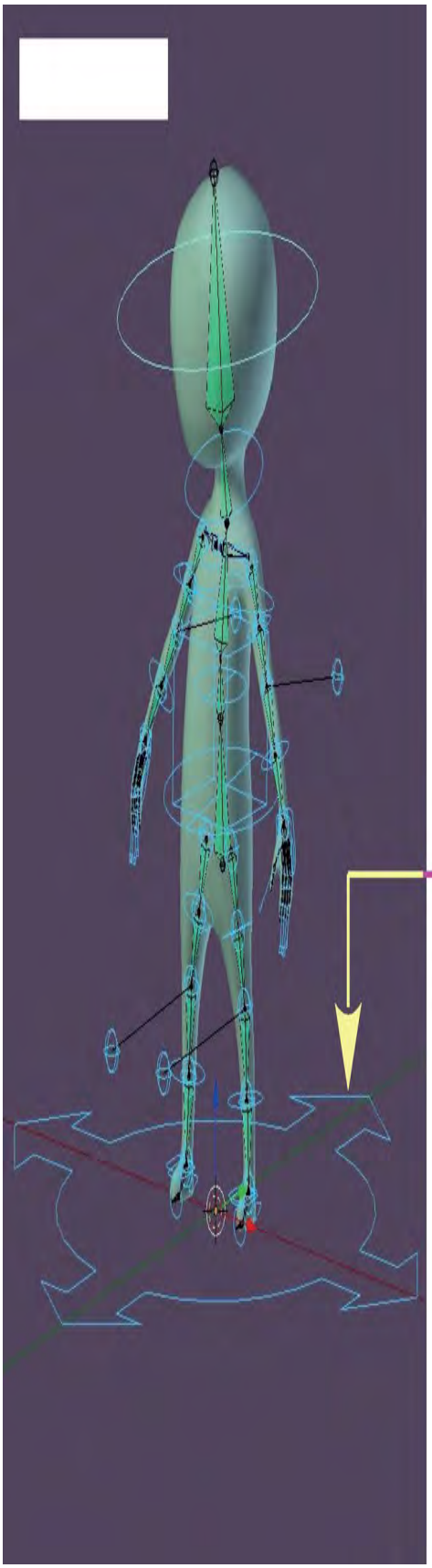


Figure 22.98

Select the Bone to be controlled then in **Pose Mode,** in the **Properties Editor, Bone Properties, Viewport Display Tab** enter the name of the new Object in the **Custom Object** panel. Adjust the Scale.



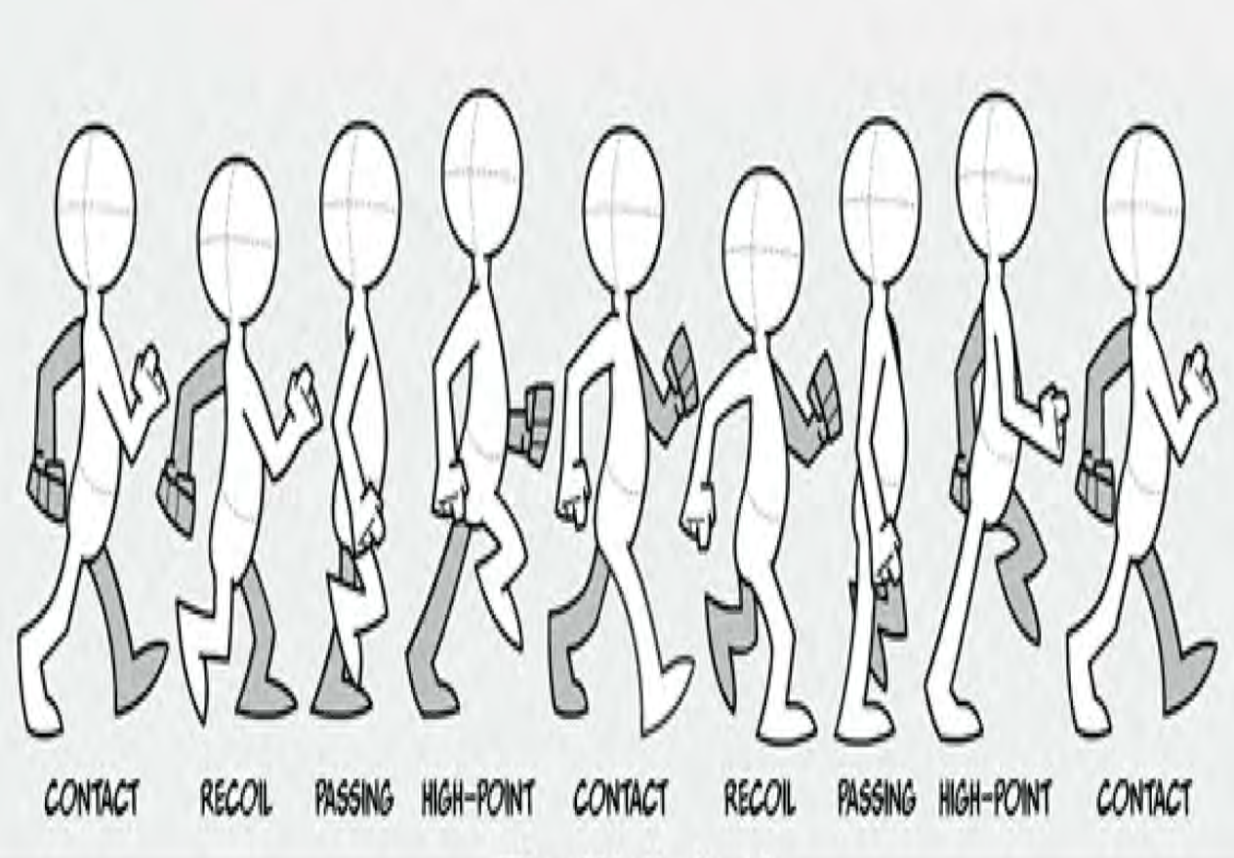
The large Control Handle at the Base of the Character is developed from the Root_Bone and is used for moving the entire Rig in the Scene.

Figure 22.99

At the beginning of the chapter it was emphasised that the instructions were to be **an introduction** to Character Rigging only. The forgoing is intended to get you started and encourage you to research detailed tutorials. The following images depict how a Character Rig may be developed to provide detailed control of Posing.

Figure 22.99 shows Chibi with a myriad of Control Bones which are more precisely named Control Handles. Each handle allows Posing of separate parts of the Character Mesh.

Using Control Handles the Character Figure is Posed at Frames in an animation creating Keyframes. This produces a Walk Cycle (Figure 22.100).



The Walk Cycle

Figure 22.100

With the Character Model animated to walk on the spot animating the Base Control Handle to follow a Path creates the illusion of the Character in the Scene.

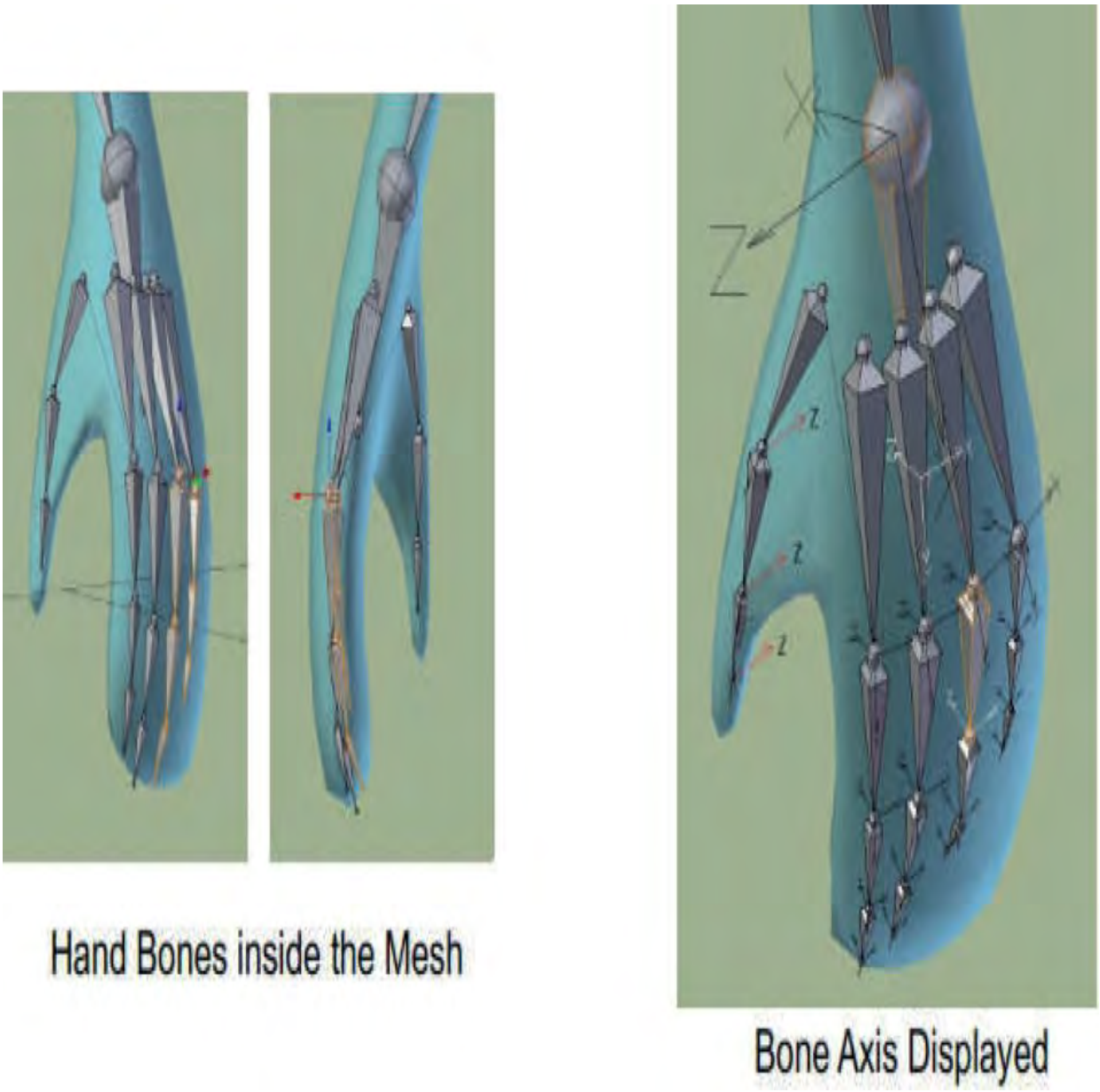


Figure 22.101

Armatures may be generated to include intricate detail such as Chibi's hands. Control Handles can be created to manipulate fingers and thumb.

22.23 Pre-Assembled Armatures

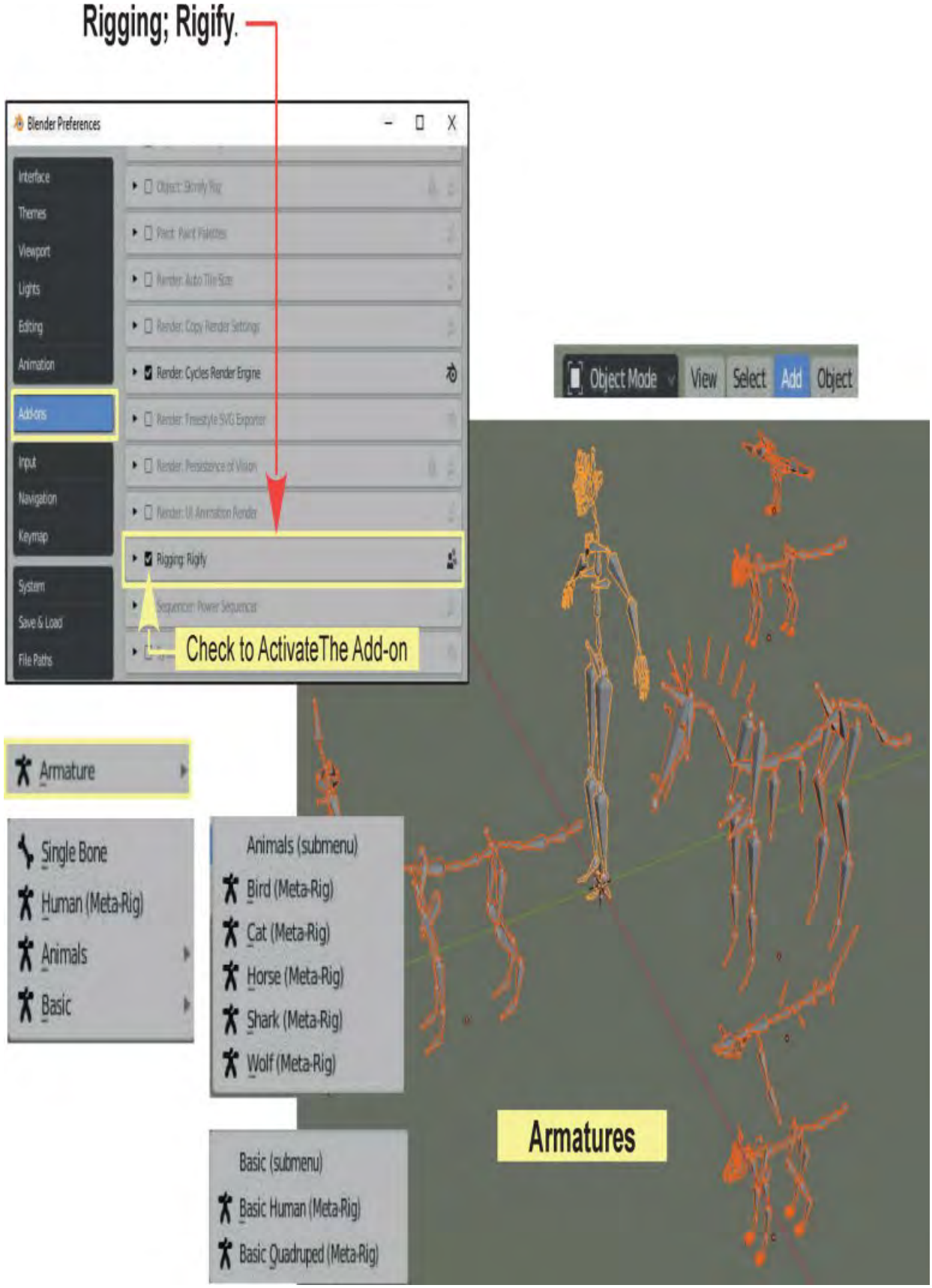

Rigging; Rigify.
Blender Preferences
Interface
Themes
Viewport
Lights
Editing
Animation
Add-ons
Input
Navigation
Keymap
System
Save & Load
File Paths
Render: Cycles Render Engine
Rigging: Rigify
Check to ActivateThe Add-on
Object Mode
View
Select
Add
Object
Armature
Single Bone
Human (Meta-Rig)
Animals
Basic
Animals (submenu)
Bird (Meta-Rig)
Cat (Meta-Rig)
Horse (Meta-Rig)
Shark (Meta-Rig)
Wolf (Meta-Rig)
Basic (submenu)
Basic Human (Meta-Rig)
Basic Quadruped (Meta-Rig)
Armatures

Figure 22.102

With the **Rigging: Rigify** Add-on activated, with the 3D Cursor in the 3D Viewport Editor, Press **Shift + A Key** and select **Armature** in the **Add Menu.**

23 Shape Keys &Action Editors

23.1 Shape Key Editor
23.2 Set Limits of Movement
23.3 Inserting Keyframes
23.4 Inserting Multiple Keyframes
23.5 The Animation
23.6 Additional Keys
23.7 Action Editor
23.8 Shape Keys and Action Editor in Practise

The **Shape Key** and **Action Editors** provide a method of quickly controlling the shape of an Object or the pose of a Character. This is accomplished by inserting **Keyframes** in an Animation Timeline.

Armature Control Handles allow the posing of the character as a whole but when detail is animated, such as facial expression or finger movement, a more refined control is desirable. If you think about an Object or a character model and all the Vertices contained in its mesh surface you will realise the impossibility of individually manipulating Vertices between the Frames of an Animation.

The **Shape Key** and **Action Editor** allow you to set up **Slider Controls** for manipulating the Mesh shape or posing the Character or Model.

The **Shape Key Editor** controls the manipulation and animation of Vertices

The **Action Editor** allows you to set up an animation of an Object's movement and scale.

23.1 Shape Key Editor

The **Shape Key Editor** allows you to control the manipulation of Vertices or Vertex Groups.

The **Shapes Key Editor** is located in the **Dope Sheet Editor.**

To demonstrate, start with the default Blender Scene, delete the Cube, and add a simple **Plane Object** which contains four Vertices. Place the Scene in **Top Orthographic View** and zoom in.

Below the 3D Viewport Editor is the **Timeline Editor**. Change the Timeline to the **Dope Sheet Editor**.
In the Dope Sheet Editor, click on the drop down in the Header where you see **Dope Sheet** and select **Shape Key Editor**.
Mode
Cache File
Mask
Grease Pencil
Shape Key Editor
Action Editor
Dope Sheet
Dope Sheet
View
Select
Marker
Channel
Key
1
10
20
30
40
50
Summary
Summary only displays if you change from the Shape Key Editor back to the Dope Sheet.

Figure 23.1

With the **Shapes Key Editor** selected, the Editor has become a simple Animation Timeline (Figure 23.2) with **Frame** numbers in the horizontal bar along the top of the Editor. There is also a vertical blue line in the Editor which is the **Timeline Cursor**.

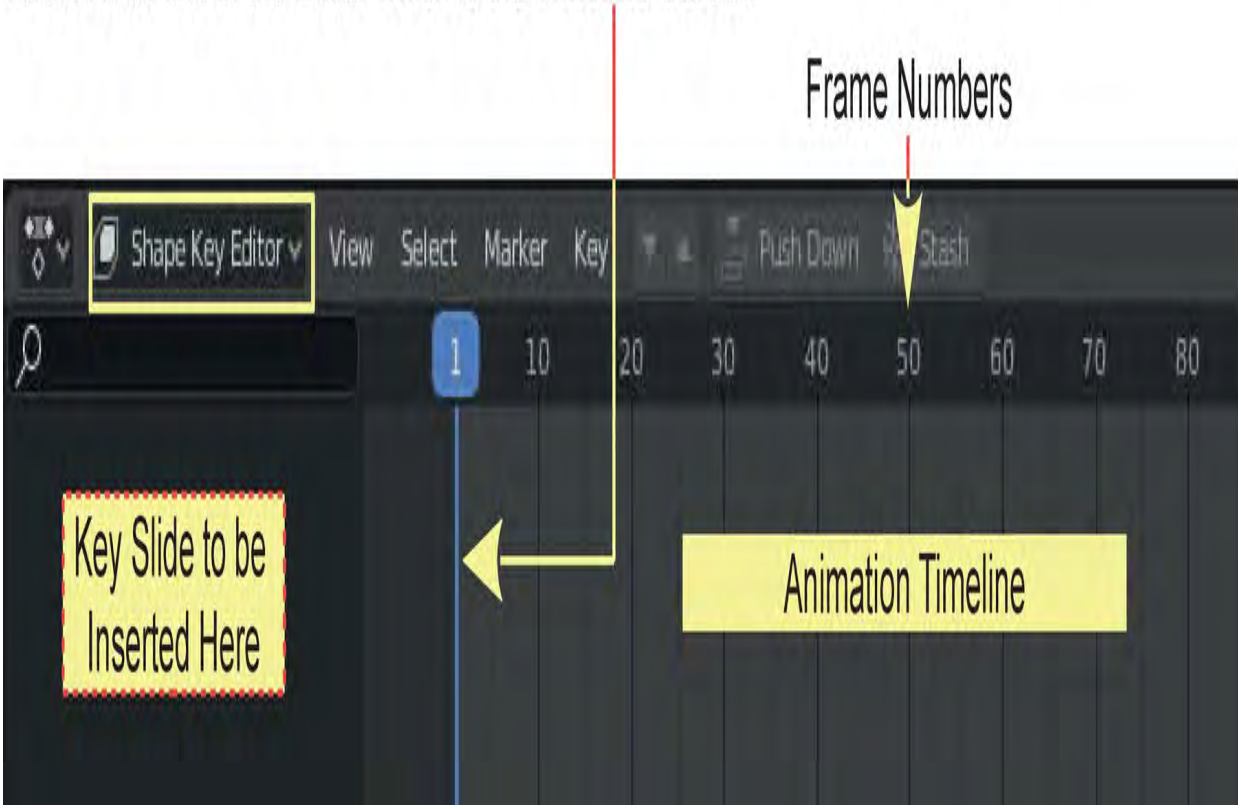


Figure 23.2

Key Sliders will be added in the **Shape Key Editor** which will allow you to control the shape of an Object in the 3D Viewport Editor. The movement of the Object's Vertices will be set within minimum and maximum limits. By moving the Vertices, via the Key Sliders, to different positions within the limits, at different Frames of the Animation Timeline you create an animation of the change in shape of the Object.

The Key Sliders to be created are applicable only to the selected Object. The first step in the process is to enter a **Basis Key in** the **Properties Editor, Object data Properties, Shape keys Tab.**

The **Basis Key** is a **Master Key** (Datablock) for subsequent Keys, for the selected Object.

Add a Basis Key

Select the Plane in the 3D Viewport Editor. In the **Properties Editor, Object Data Properties, Shape Keys Tab** click on the **Plus sign** (Figure 23.3).

Properties Editor

The Tab expands, showing a **Basis Key** inserted.

Object Data Properties

Figure 23.3

Figure 23.4

Add a Key Slider

In the **Properties Editor, Shape Keys Tab** click on the plus sign again and **Key 1** will be added.

Figure 23.5

Figure 23.6

23.2 Set Limits of Movement

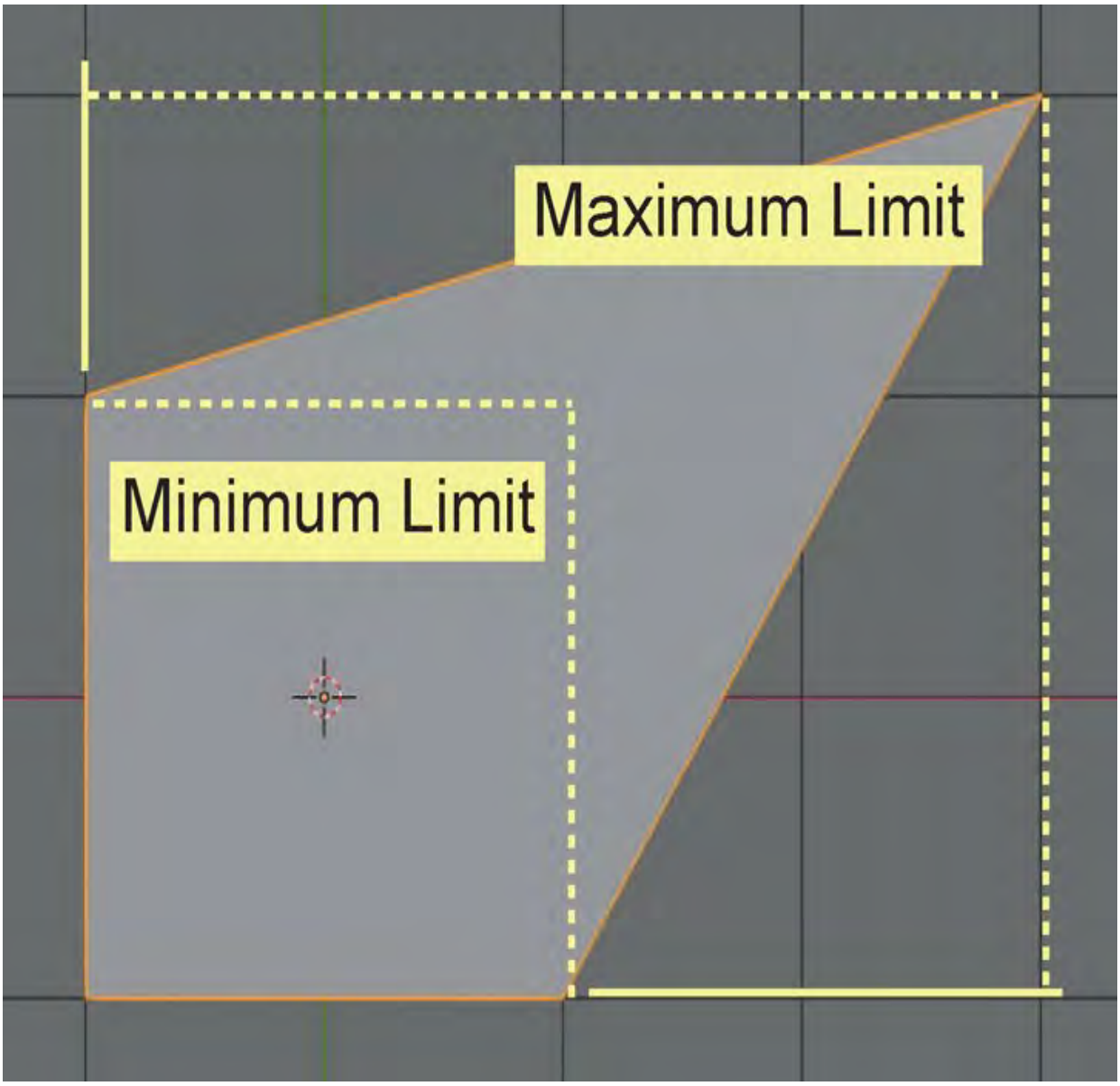


Figure 23.7

With a Key Slider in place you set limits for what is to be controlled.

With the Mouse Cursor in the 3D Viewport Editor, in Top Orthographic View, **Tab to Edit Mode.** Deselect the Vertices (LMB Click in the Viewport). Select a single Vertex and drag it (press the G Key and drag the Mouse) to where you want it to move to (maximum movement Limit - Figure 23.7). Tab back to **Object Mode.** The Vertex reverts to its original position. Moving the vertex in Edit Mode has **set the limits** for the movement.

With the 3D Viewport Editor In Object Mode drag the **Key 1 Slider** in the Dope Sheet, Shape Key Editor all the way to the right (0.000 — 1.000 Figure 23.8) then return to 0.000. The Plane changes shape in the 3D Viewport Editor.

Figure 23.8

23.3 Inserting Keyframes

Dragging the slider in the **Shape Key Editor** and returning it to 0.000 automatically sets a **Keyframe** in the **Timeline.** The Keyframe is placed at the location of the Timeline Cursor (the blue line at Frame 1) and displays as little orange diamonds (Figure 23.8). The keyframes display in the Shape Key Editor and in the Timeline Editor.

Note: Dragging the slider moves the selected Vertex only within the limits that were set. The slider value is from 0.000 to 1.000, that is from the initial position to the maximum limit of the movement.

Inserting a Second Keyframe

To insert a second Keyframe move the blue line **Cursor** in the **Shapes Key Editor** to another Frame (Frame 50). Move **Key 1 Slider** until the Vertex in the 3D Viewport Editor is where you want it (0.759). Release the mouse button (Figure 23.8). Leave the Slider at the chosen value of the second Keyframe. Return the blue line Cursor to Frame 1.

When you drag the blue line Cursor, the shape of the Plane in the 3D window changes with the corner (Vertex) moving from its initial rest position to where you positioned it at Frame 50.

Maybe you didn't get the position of the Vertex exactly where you intended. Dragging the Key Slider is a bit touchy when you want an exact location.

Deleting a Keyframe

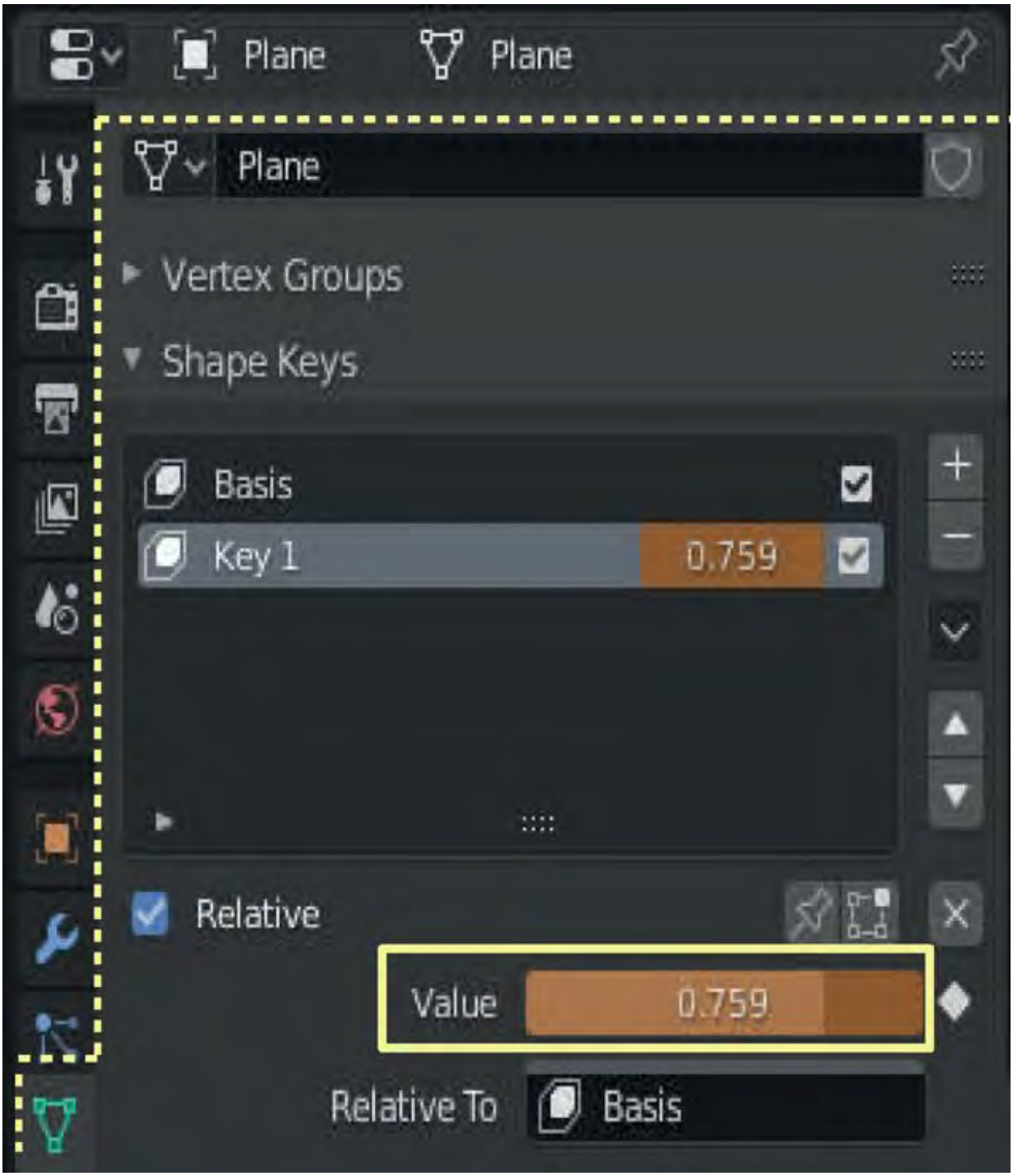


Figure 23.9

To delete a Keyframe and start over, place the blue Cursor in the Shape Key Editor at the Frame in the Timeline where you want to remove

Keyframes. **RMB click on Value bar** (Figure 23.9) in the **Properties Editor, Object Data Properties, Shape Keys Tab** and select **Delete Keyframe.**

Note: The relative Value Bar replicates the Key 1 Slider.

Add a New Keyframe

You may add a new Keyframe by moving the Key 1 slider again. If you want an exact value double click the slider and type in a value.

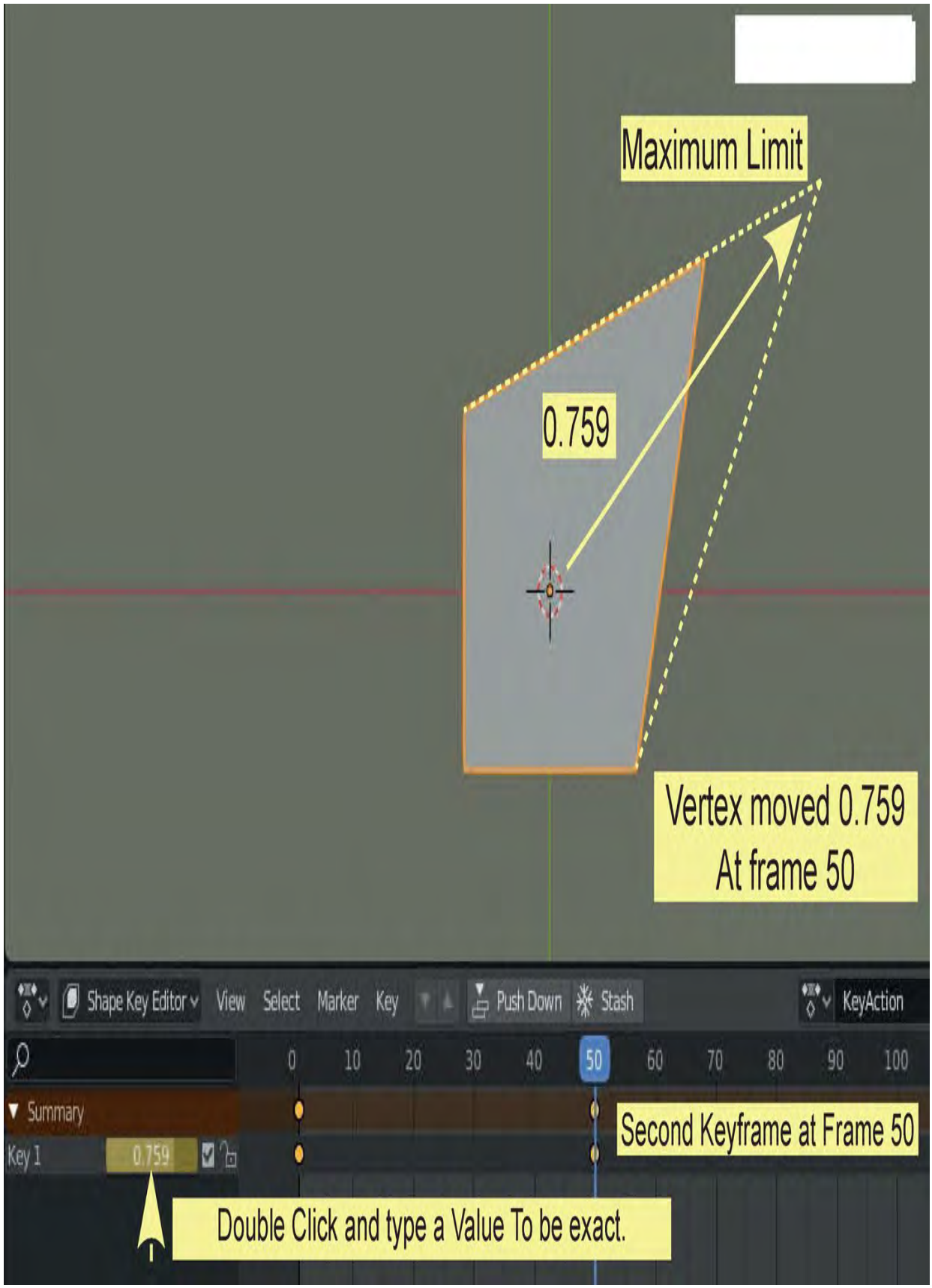

Maximum Limit
0.759
Vertex moved 0.759
At frame 50
Shape Key Editor
View
Select
Marker
Key
Push Down
Stash
KeyAction
0
10
20
30
40
50
60
70
80
90
100
Summary
Key 1
0.759
Second Keyframe at Frame 50
Double Click and type a Value To be exact.

Figure 23.10

23.4 Inserting Multiple Keyframes

After placing a Keyframe you do not have to return the Timeline Cursor to frame 1. Move it to another Frame. Move the Key 1 Slider. Another Keyframe is added. Repeat the process for multiple Keyframes (Figure 23.11).

Figure 23.11

23.5 The Animation

To this point Keyframes have been added in the **Shape Key Editor Timeline.** You may scrub the Timeline (drag the blue line Cursor) to see the shape change in the 3D Viewport Editor. To see an animation play open the **Timeline Editor** and press the **Play** button (Figure 23.12).

Figure 23.12

23.6 Additional Keys

Additional **Keys** are added to the Dope Sheet Summary for Animating other parts of the Mesh (other Vertices).

Move the cursor in the **Shapes Key Editor Timeline** to Frame 1. In the **Properties Editor, Object Data Properties, Shape Key Tab,** click on the plus sign again to add **Key 2** (Figure 23.13). In the 3D Viewport Editor, Tab to Edit Mode, select a different Vertex, and move it somewhere to set the limit of movement. Tab back to Object Mode and you'll see that Key 2 has been

added to the Dope Sheet Summary (Figure 23.13). Repeat the Keyframing process using Key 2 for the new Vertex.

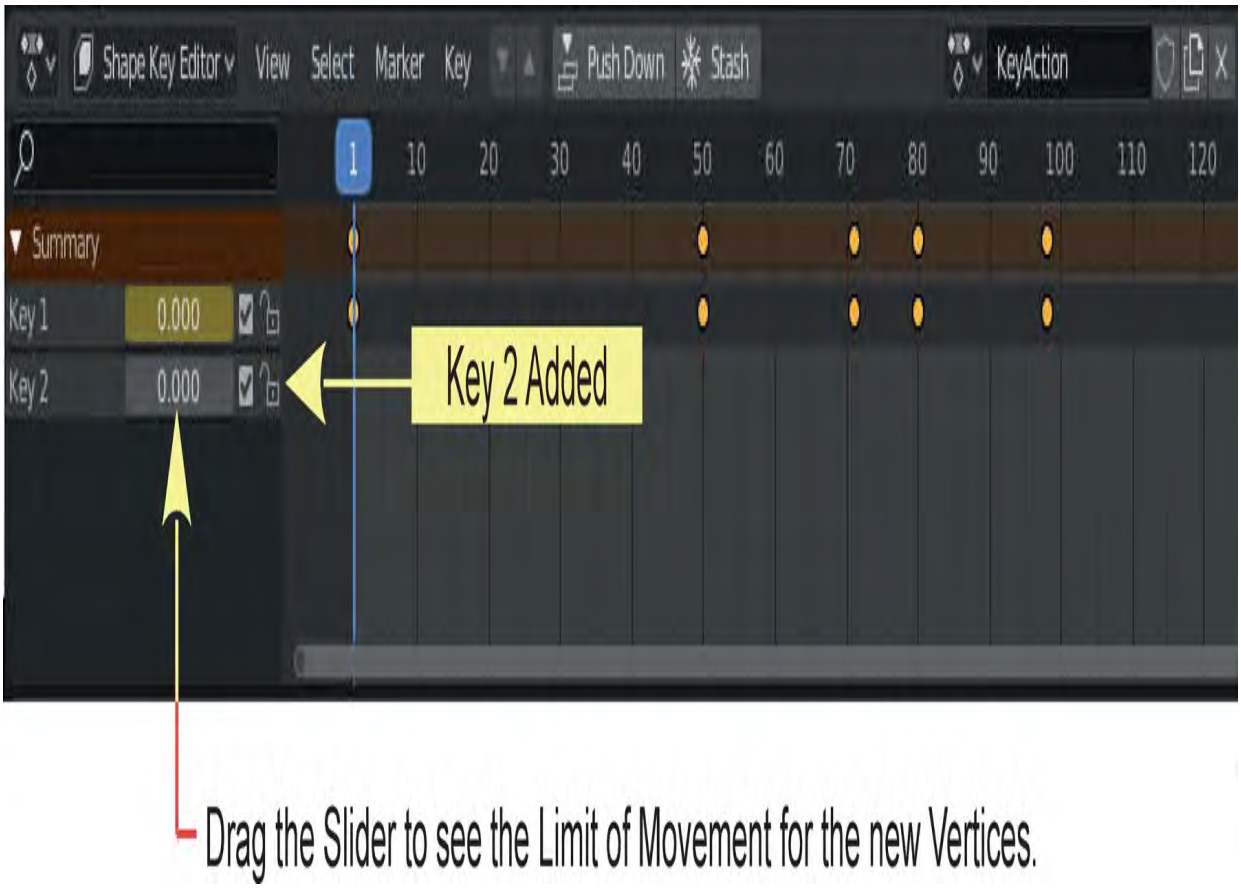


Figure 23.13

After inserting Keyframes for the new Vertex controlled by Key 2, scrubbing the Timeline Editor Cursor or playing the Animation will show both Vertices moving as the animation plays.

23.7 Action Editor

Figure 23.14

The **Action Editor** is used to control the Translation, Rotation and Scale of the Object as a whole. **Select the Object in the 3D Viewport Editor.**

In the **3D Viewport Editor** select the default **Cube.** With the Mouse Cursor in the 3D Viewport Editor press the **I Key** and select **Location, Rotation & Scale.** This inserts a **Keyframe at Frame 1** and enters an **Object Transforms Summary** in the Action Editor. Click on the triangle preceding **Object Transforms** to display the Keyframe entries for **X,Y and Z, Location Rotation and Scale** (Figure 23.15).

Figure 23.15

In the **Action Editor Header,** click on **View** and check **Show Sliders;** sliders display for each Keyframe component (Figure 23.16).

Figure 23.16

By repositioning the Cursor (blue line) in the Action Editor to a new Frame and moving the Sliders, you manipulate the Cube in the 3D Viewport Editor (Figure 23.17). After moving the Slider, **RMB click** on the new value and select **Replace Keyframe.** When the Cursor is positioned at a different Frame and slider values are changed, Keyframes are inserted, which produces an Animation.

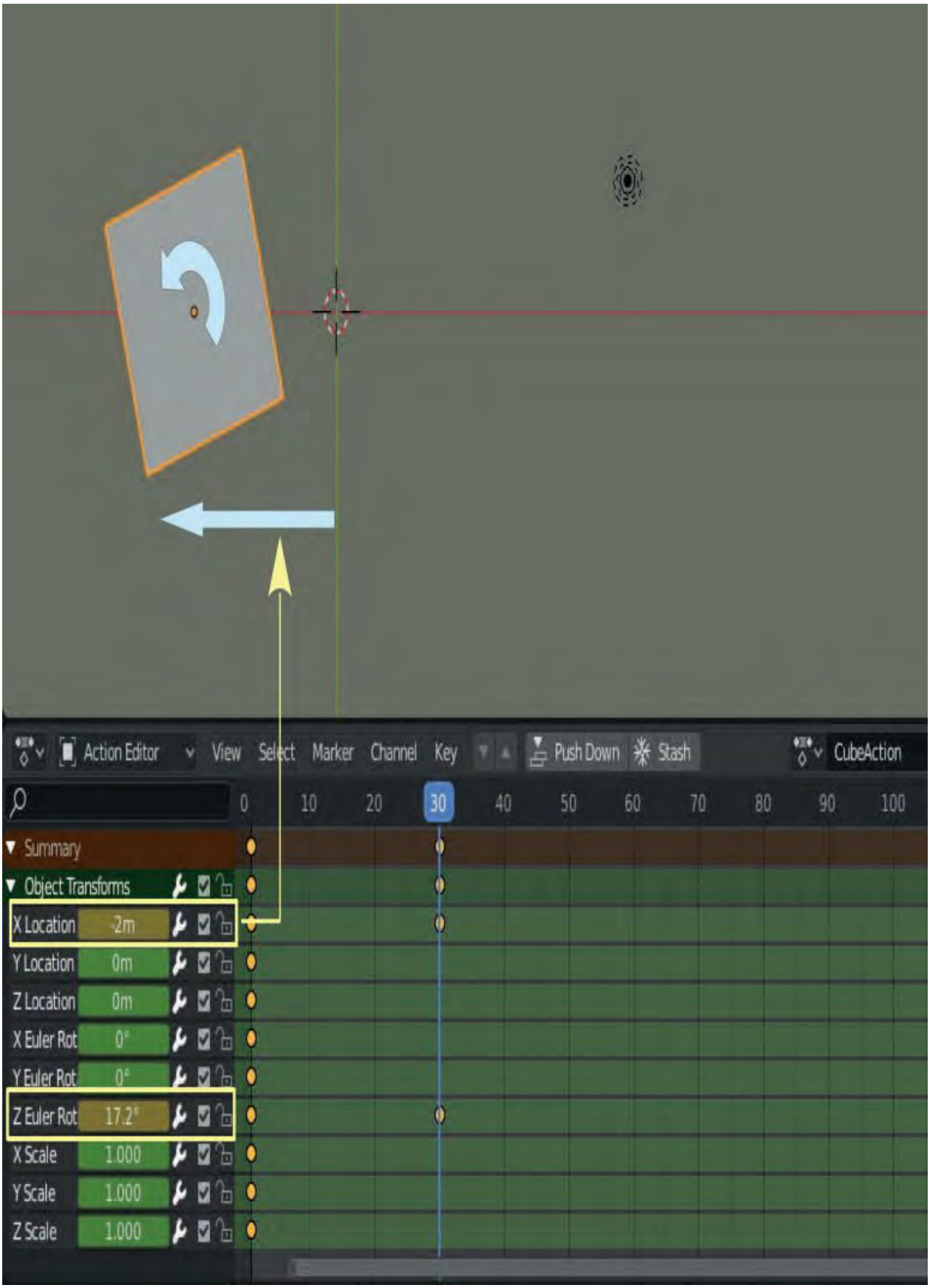

Action Editor
View
Select
Marker
Channel
Key
Push Down
Stash
CubeAction
0
10
20
30
40
50
60
70
80
90
100
Summary
Object Transforms
X Location -2m
Y Location 0m
Z Location 0m
X Euler Rot 0°
Y Euler Rot 0°
Z Euler Rot 17.2°
X Scale 1.000
Y Scale 1.000
Z Scale 1.000

Figure 23.17

23.8 Shape Keys and Action Editor in Practise

The forgoing examples show you the fundamentals of the tools but they are not very exciting and you could be left wondering what to do with them in some practical application. To expand on the topics perform the following exercise:

In a new Blender Scene delete the **Cube** and add a **Monkey Object.** Place the Scene in **Front Orthographic view** and zoom in to fill the 3D Viewport Editor with Suzanne's head. **Tab to Edit Mode** and select the Vertices in the face as shown in Figure 23.18.

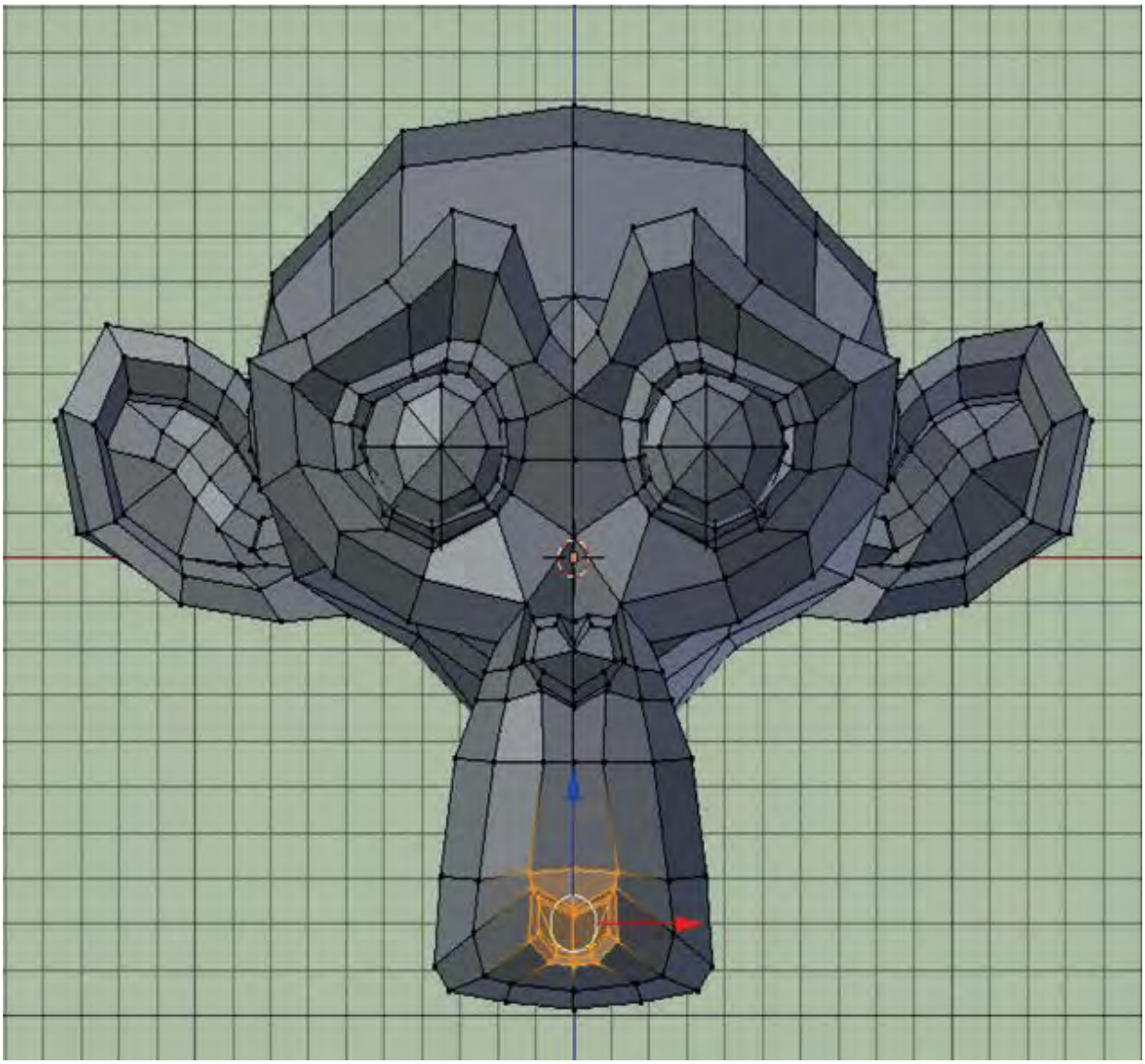


Figure 23.18

Change the Timeline Editor to the **Dope Sheet Editor** then to the **Shape Key Editor Mode.** In the **3D Viewport Editor** position Monkey to have the mouth visible (Figure 23.18 -19).

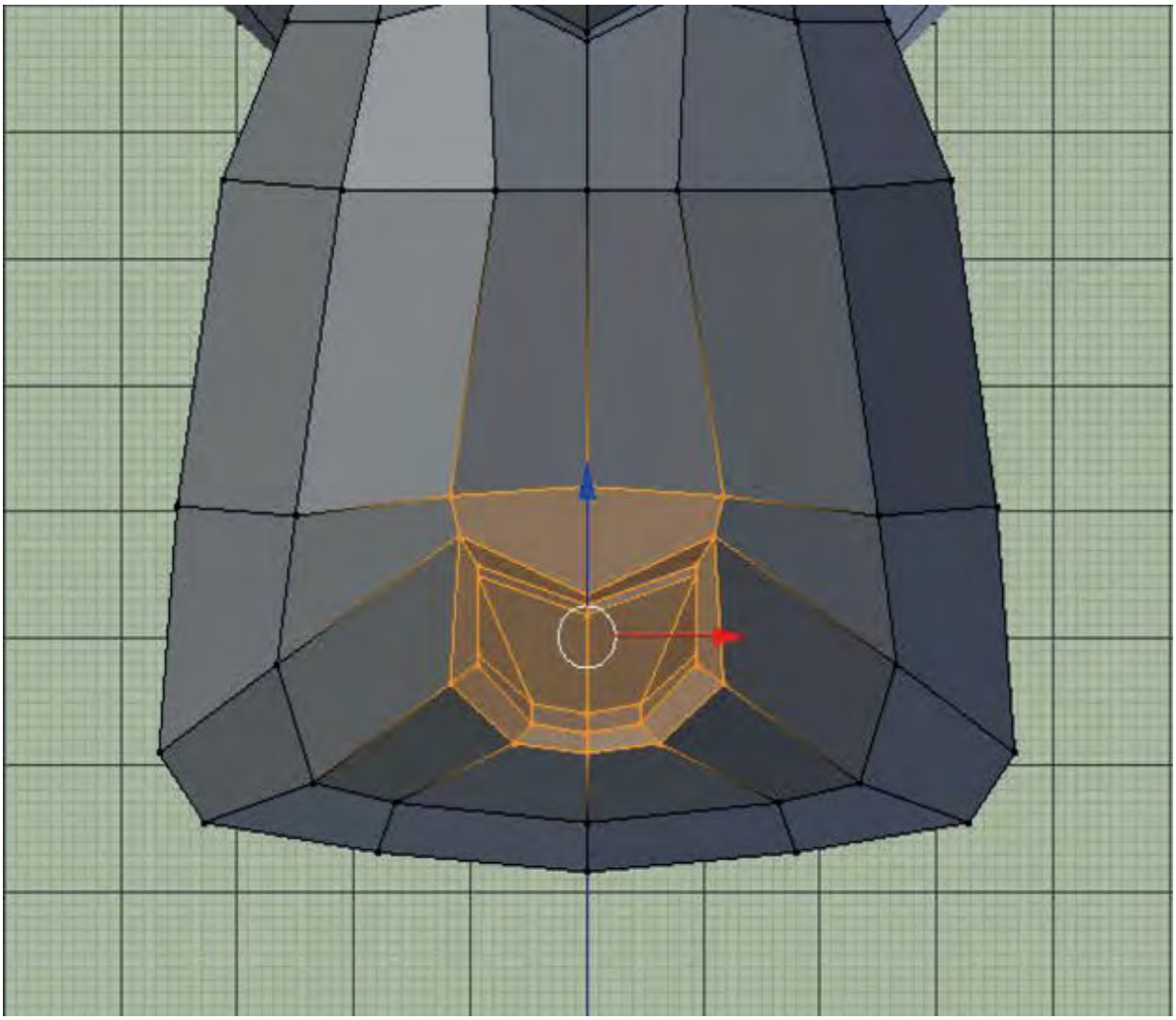


Figure 23.19

Tab to Object mode.

In the **Properties Editor, Object Data Properties, Shape Key Tab** click on the plus sign to add a **Basis Key.** In the **Action Editor Header** click **New** to enter a **Summary** in the Dope Sheet. Click the plus sign again in the Properties Editor, Object Data Properties Shape Keys Tab to insert **Key 1.**

You are about to make Suzanne laugh. Suzanne's lips will move but you will have to use your imagination as far as sound is concerned.

In the **3D Viewport Editor** change to **Right Orthographic View .** Tab to **Edit Mode.** Use the widget and move the selected Vertices to the left making

Suzanne's lips protrude slightly (Figure 23.20). Change to **Front Orthographic View .** Scale the selected Vertices up on the Z Axis and a little bit on the X Axis (Figure 23.21). Using the widget move the Vertices up. **Tab to Object Mode.** The Vertices revert to their original location.

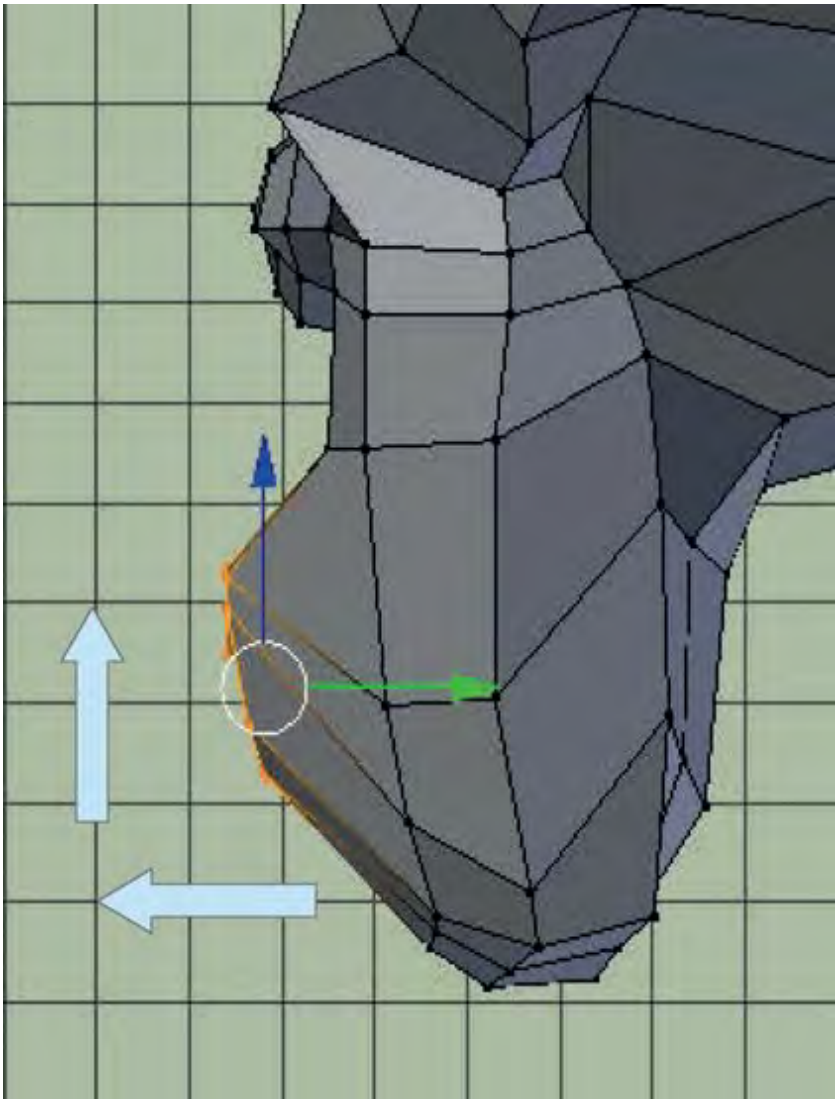


Figure 23.20

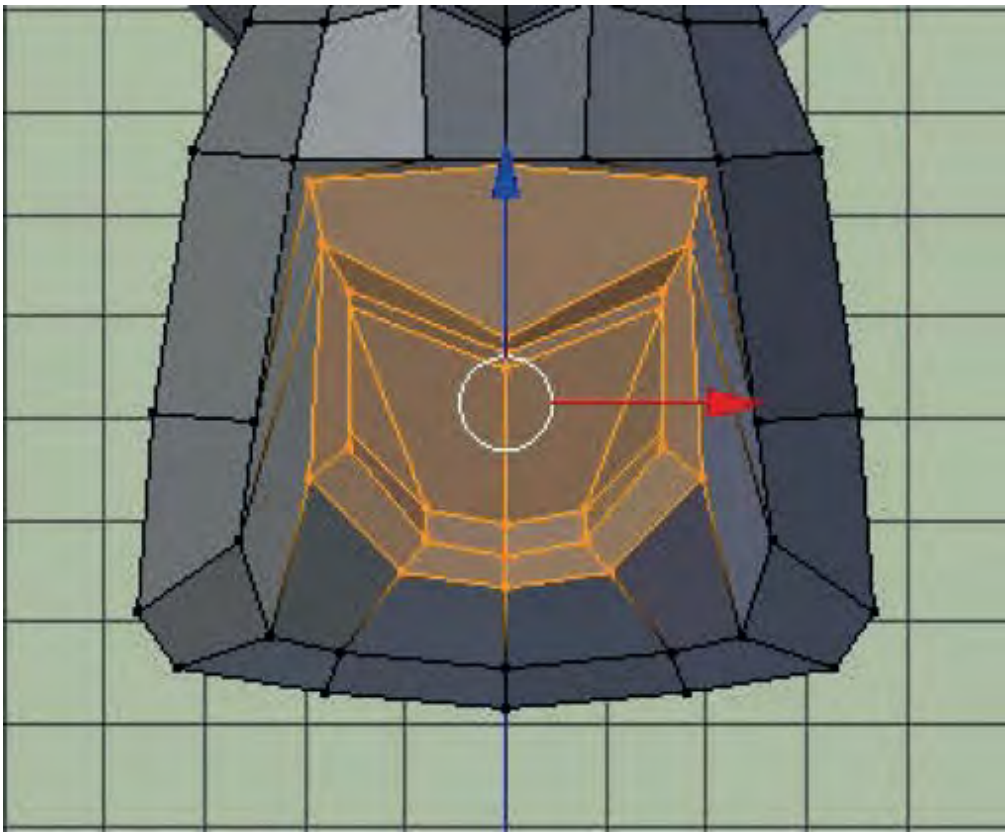


Figure 23.21

In performing the scaling and location operations you have set the limits of movement for **Key 1** for each of the Vertices that were selected.

Change the Dope Sheet Action Editor to Shape Key Editor. In the Header click **View** and check **Show Sliders.**

Expand the Summary and Move the Key 1 Slider to see Suzanne's mouth move. Remember, moving the Slider inserts a Keyframe at the Frame number in the Animation Timeline where the Timeline Cursor is positioned (Frame 1) (Figure 23.22).

Figure 23.22

Figure 23.23

If you don't want Suzanne to start laughing at the start of the animation move the cursor down the track in the Timeline before moving the Slider

(Frame 30 – Figure 23.23).

Remember; by default, Blender has a 250 Frame animation set in the Timeline. If you place the Shape Key Editor Cursor beyond this it will have no effect unless you change the End Frame value in the Timeline Editor Header.

Place a series of Keyframes in the animation. Move the Cursor to Frame 40. Move the Key slider leaving it in position. This inserts a Keyframe at frame 40. Move the Cursor to frame 50. Move the slider. Repeat the operation as many times as you like (Figure 23.24).

In the Timeline Editor, go back to Frame 1 and play the animation. Monkey's mouth moves as the animation plays.

Figure 23.24

To create more action in the Animation have Suzanne turn her head as she laughs.

Zoom out on the 3D Viewport Editor. Change to User Perspective View. In the Action Editor Header click the New button. With the Timeline Cursor at

Frame 1, Press I Key and select Location, Rotation & Scale to place action Keyframes (Figure 23.25).

Action Editor
View
Select
Marker
Channel
Key
Push Down
Stash
Action.001
1
10
20
30
40
50
60
70
80
90
100
110
120
Summary
Object Transforms
X Location
0 m
Y Location
0 m
Z Location
0 m
X Euler Rota
0°
Y Euler Rota
0°
Z Euler Rota
0°
X Scale
1.000
Y Scale
1.000
Z Scale
1.000
Playback
Keying
View
Marker
Summary
Suzanne
Action.001
Object Transforms
X Locatic
0 m
Y Locatic
0 m
Z Locatic
0 m
X Euler R
0°
Y Euler R
0°
Z Euler R
0°
X Scale
1.000
Y Scale
1.000
Z Scale
1.000
Key

Figure 23.25

Move the Cursor to Frame 100 to coincide with the end of the **Shape Key** animation and move the **Z Euler Rotation slider to 45 degrees** (Rotation in Blender is measured in Euler units).

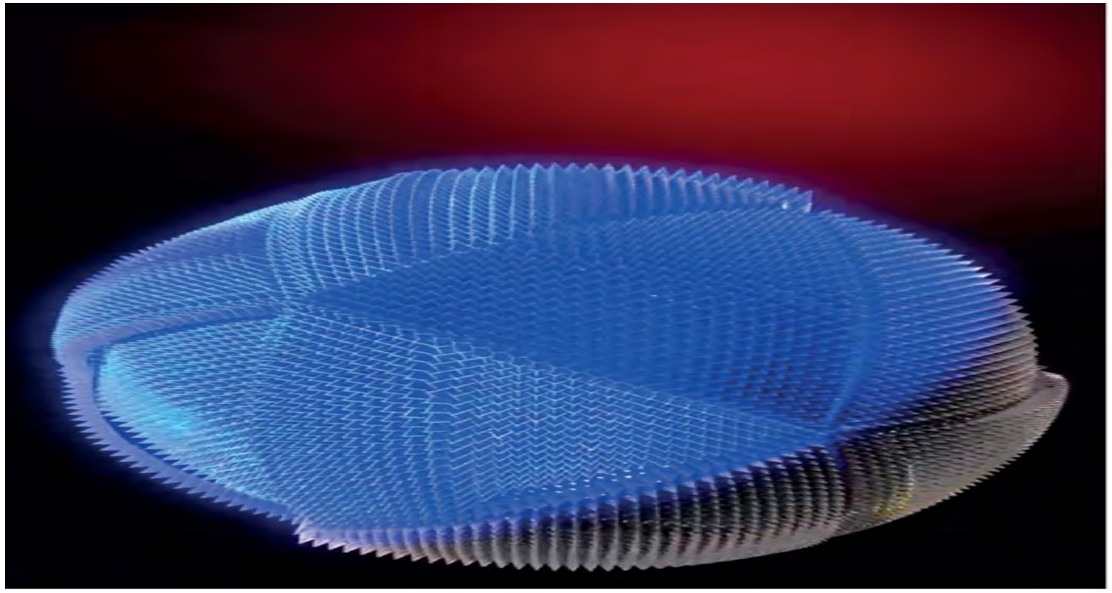
Place the 3D Viewport Editor in Camera View and play the animation. Suzanne's mouth moves while turning to face the Camera.

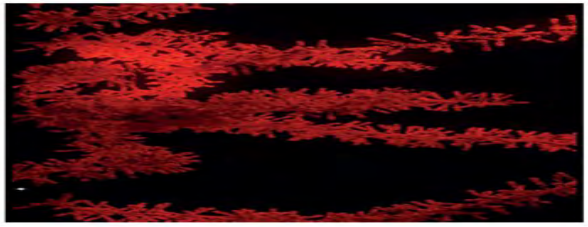
This has been a very simple practical demonstration so use your imagination and experiment.

24 Particle Systems

24.1 The Default Particle System
24.2 The Emissions Tab
24.3 The Source Tab
24.4 The Cache Tab
24.5 The Velocity Tab
24.6 Particle Display
24.7 Particle Emission Options
24.8 Order of Emission
24.9 Normals
24.10 Particle Modifiers
24.11 Particles Array
24.12 The Viewport Display Tab
24.13 Particle Interaction
24.14 Wind Force Effect
24.15 Boids Particles
24.16 Hair Particles
24.17 Particles Edit Mode
24.18 Particles for Arrays
24.19 More Arrays
24.20 The Assignment Panel
24.21 Particle Exercises
24.22 Multiple Particle Systems
24.23 Keyed Particles

Particle Systems are used to simulate effects like fire, dust, clouds and smoke and for creating hair, fur, grass and other strand based Objects. When a Particle System is applied to an Object it causes the Object to generate and emit Particles which may be configured to display in a multitude of ways creating static and animated patterns or as Models of Characters. For example, an army of solders or a swarm of insects.



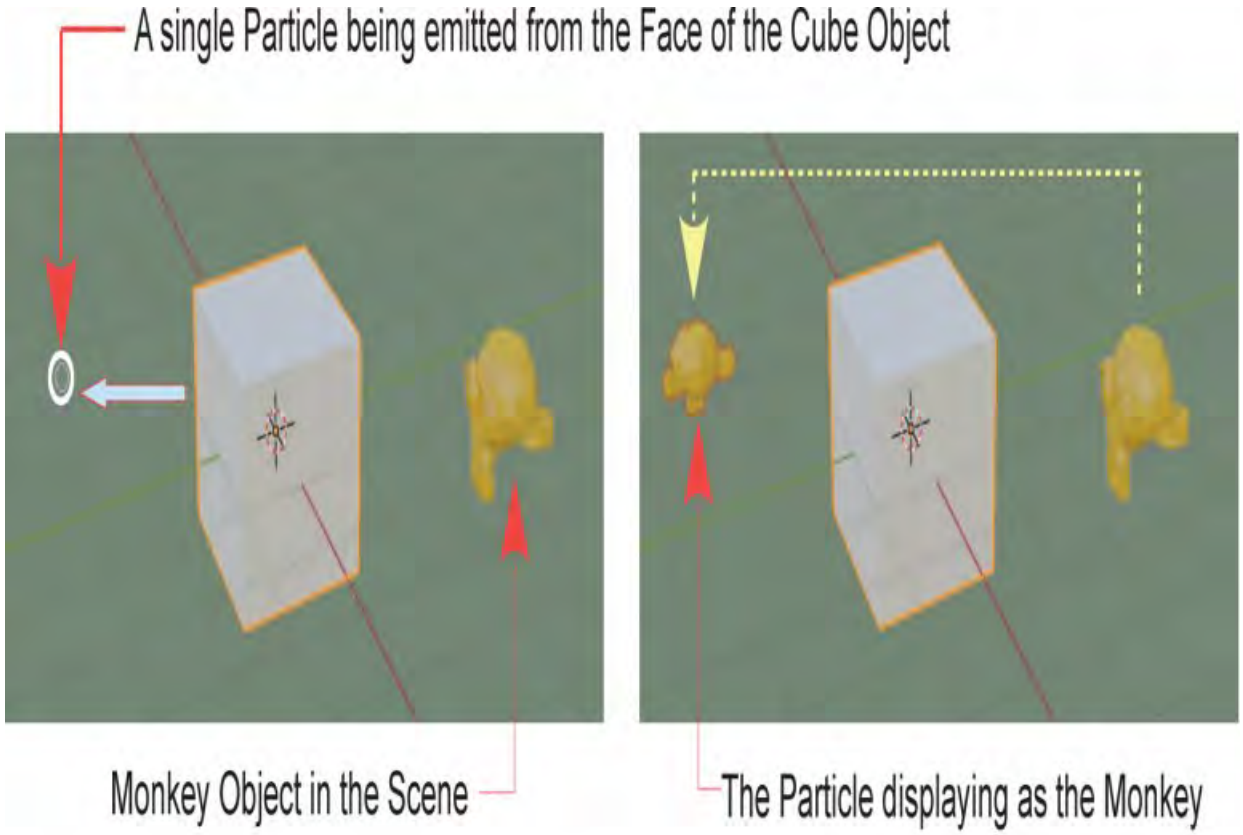


The preceding examples are screen captures from a YouTube video created by:

In Blender **Particles,** by default, appear as points or small circles on the computer screen, being emitted from an Object. To Emit Particles from an Object a **Particle System** is assigned (added) to the Object then an animation sequence is run.

Adding a Particle System to an Object creates a system with default settings which is ready to run by itself. To create Particle effects you modify the settings.

Particles are simply points on the computer Screen which can be configured to display in different ways. The Particles (points) do not render when the Scene is Rendered. The Particles are reconfigured to display as other Objects in the Scene which will Render.



Render of Camera View
Before reconfiguring the Particle
To display as the Monkey

Rendered of Camera View
After configuring the Particle
To display as the Monkey

Figure 24.1

Usually a Particle System will be used to generate more than one Particle. In fact, it is usual to create thousands of Particles and possibly from multiple Objects in a Scene.

To see how this is accomplished follow a few simple instructions and run a Particle System.

24.1 The Default Particle System

To set up a default **Particle System** open a new Scene in Blender. Delete the default Cube Object and add a **UV Sphere**.

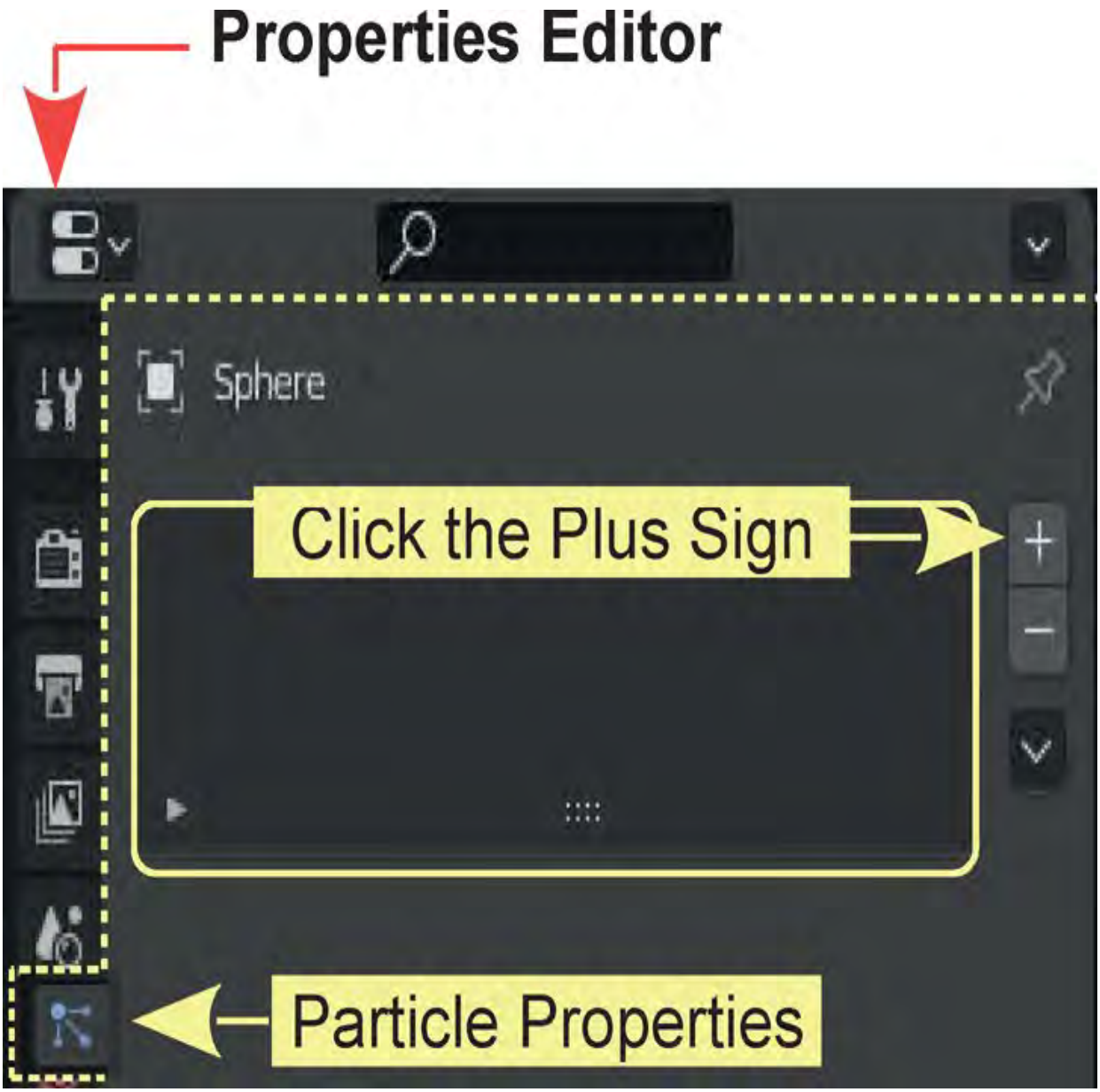


Figure 24.2

Particles are emitted from the Vertices, the Faces or from the Volume of a mesh. Using a UV Sphere provides a reasonable number of Vertices and Faces from which to emit the Particles. Leave the default values for the Sphere as they appear in the Properties Editor. With the UV Sphere selected, go to the **Properties Editor, Particles Properties**. Click on the **plus sign** to add a **Particle System** (Figure 24.2).

Sphere
ParticleSystem
ParticleSystem
ParticleSettings
Emitter
Hair
Emission
Number
1000
Seed
0
Frame Start
1.000
End
200.000
Lifetime
50.000
Lifetime Randomn...
0.000
Source
Cache
Velocity
Rotation
Physics
Render
Viewport Display
Children
Field Weights
Force Field Settings
Vertex Groups
Textures
Custom Properties
Tab
Dimpled Area
Tabs
Tab Closed
Tab Open

Figure 24.3

The **Particles Properties** open displaying the **Tabs** (panels) that control the system. Blender has automatically created a default Particle System for the UV Sphere (Figure 24.3).

Note:

When a Particle System is created it is unique to the selected Object in the 3D Viewport Editor.

The arrangement of the **Tabs** (panels) in the **Properties Editor** is purely a matter of convenience. There is no order of priority. The Properties Editor is arranged with the Tabs in a vertical stack at the LHS of the panel. The Tabs (panels) may be moved up or down by clicking and dragging the dimpled area in the upper RH corner of each Tab.

Particles only display in the 3D Viewport Editor when an animation sequence is run by activating Play in the Timeline Editor or scrubbing (dragging) the Timeline cursor (vertical blue line) to a Frame in the animation.

The default Particles display as small white sphers.

Note the default **Type: Emitter** under **Particle Settings**.

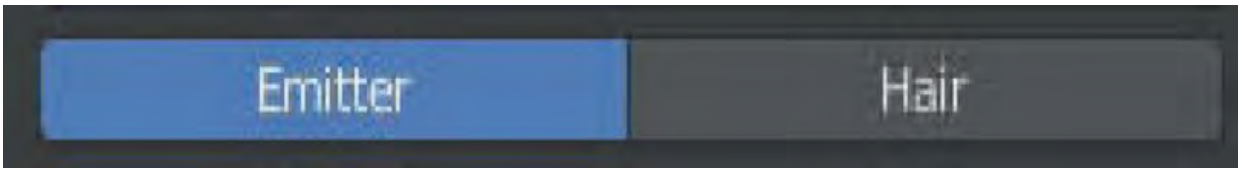


Figure 24.4

The alternative system is **Hair** (Figure 24.4). Type: Hair is a unique system which will be discussed later in this chapter.

To see the default **Particle System** in action, press the **Play button** in the **Timeline Editor**. This runs an animation sequence showing Particles being generated (Figure 24.5).

The animation plays for 250 frames then repeats. Press **Esc** to stop the animation. Advance the animation to Frame 63 by pressing the right arrows on the Keyboard (**with the Mouse Cursor in the Timeline Editor**) or by clicking RMB, holding and dragging the blue line Cursor in the Timeline Editor or by RMB clicking on frame 63 in the Timeline Editor. The Particles will be displayed as they occur at Frame 63.

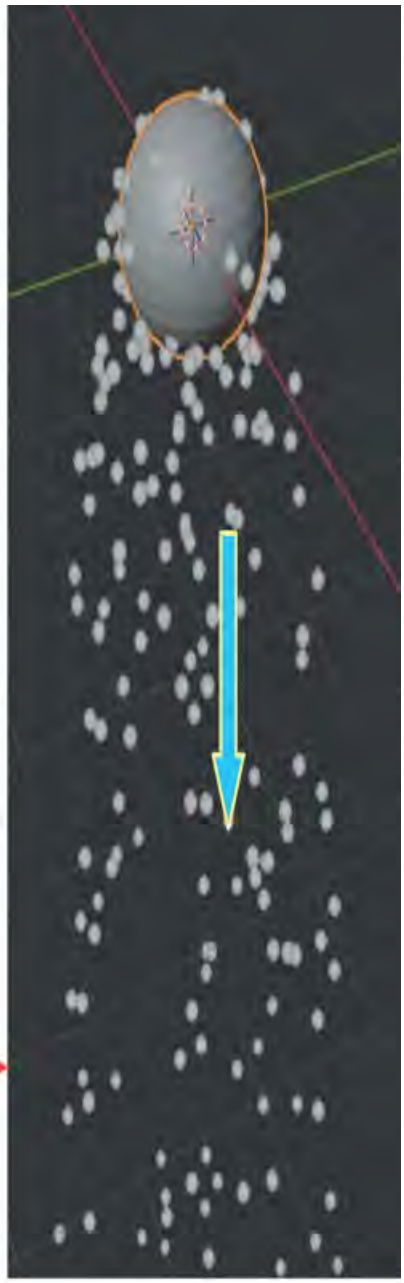


Figure 24.5

Note: The **Timeline Editor** is displayed across the bottom of the Screen (Figure 24.6). The blue line (Timeline Cursor) moves as the animation plays. With the Emitter Object selected (the UV Sphere), the animation will play showing Particles as small white spheres being emitted and falling towards the bottom of the Screen.

The example has demonstrated a simple Particle System being applied. The Particles emitted from the UV Sphere cascade down; this occurred since there is a **gravitational effect** applied (see the **Scene Properties** in the Properties Editor).

In the **Properties Editor, Scene Properties, Gravity Tab,** click on the tick next to **Gravity** (removes the tick) to remove the gravitational effect.

Set the animation in the Timeline Editor back to Frame 1 and replay the animation.

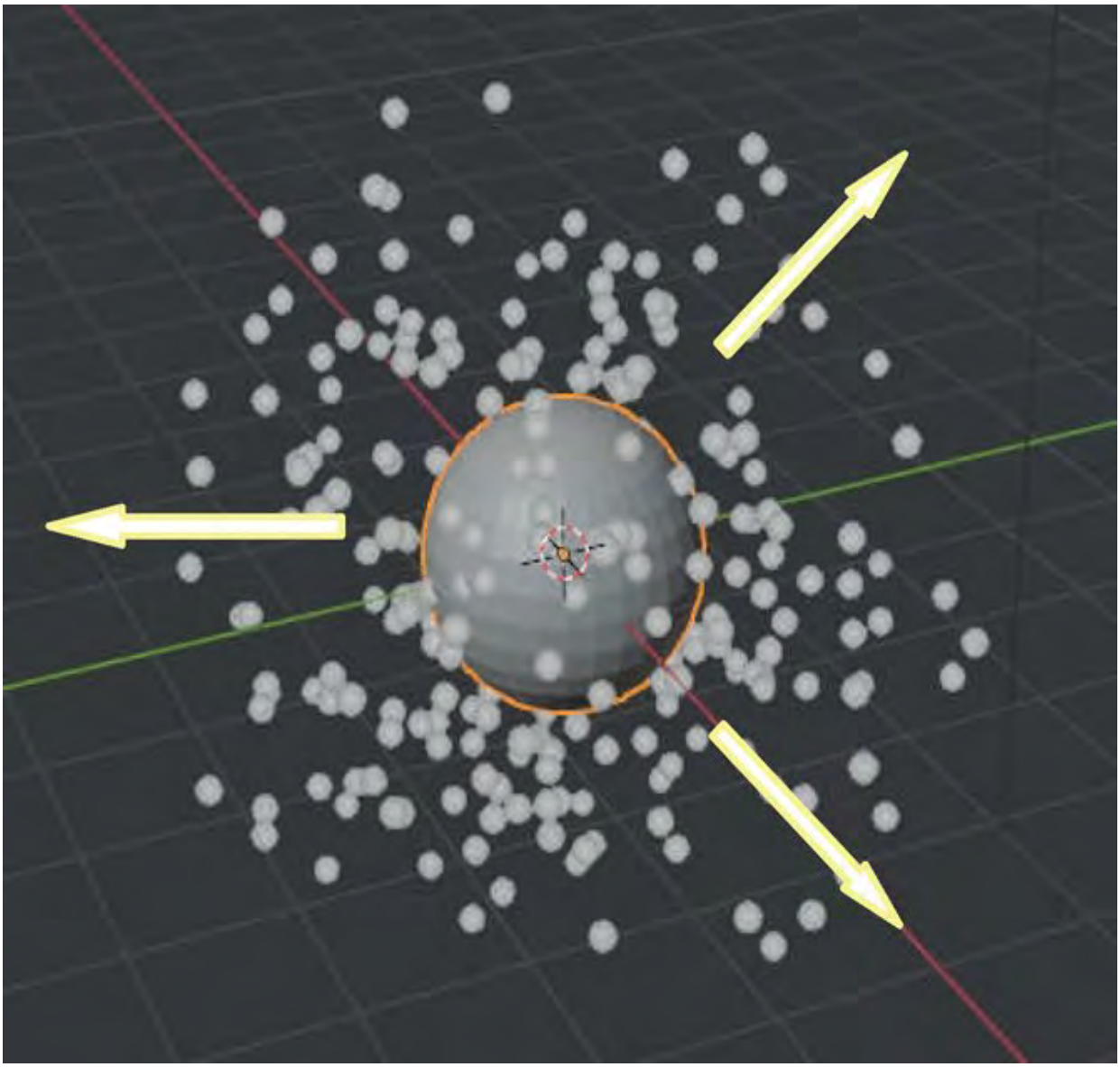


Figure 24.7

Replaying the animation shows Particles Emitted from the UV Sphere disperse in all directions away from the Sphere (Figure 24.7).

Note: The Particles move for a certain time and disappear before the end of the animation. The time that the Particles display is set in the **Emission Tab (Lifetime**Figure 24.8).

24.2 The Emissions Tab

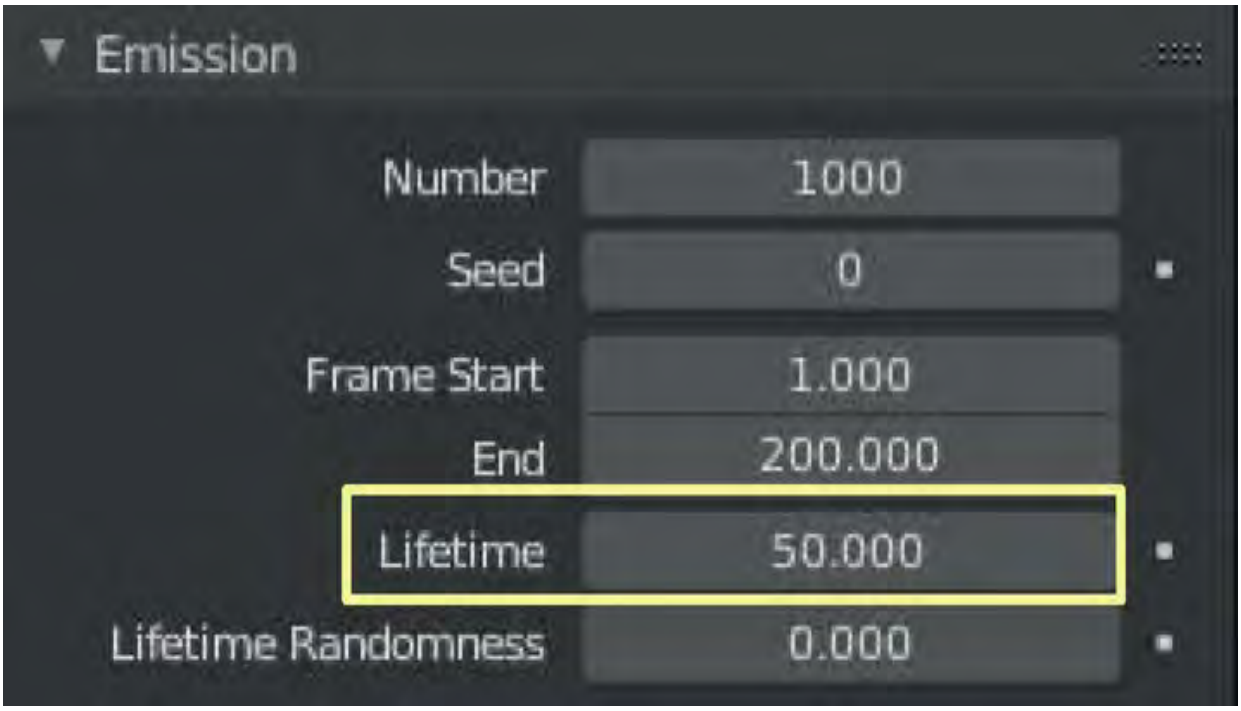


Figure 24.8

You control the Particle display in the 3D Viewport Editor in the **Particle Properties, Emissions Tab** by adjusting values (Figure 24.8).

Number: The total number of Particles to be Emitted over the length of the animation.

Seed: Sets a randomized **Emission** variation.

Frame Start: The Frame number in the animation to start emitting.

End: The Frame number to Particles.

Lifetime: The number of Frames in the animation that Particles, **which have been Emitted,** will display for. The default number of Frames is 50.

Lifetime Randomness: Gives **Lifetime** a random variation.

With the default settings, 1000 Particles will be emitted over the length of the animation. The default animation length is 250 Frames (see the Timeline Editor). The Particles will begin Emitting at Frame 1 and end at Frame 200. The Particles display for a Lifetime of 50 Frames, therefore, the last Particle

to be emitted (at Frame 200) displays for 50 frames, that is, to the end of the animation.

24.3 The Emission Source Tab

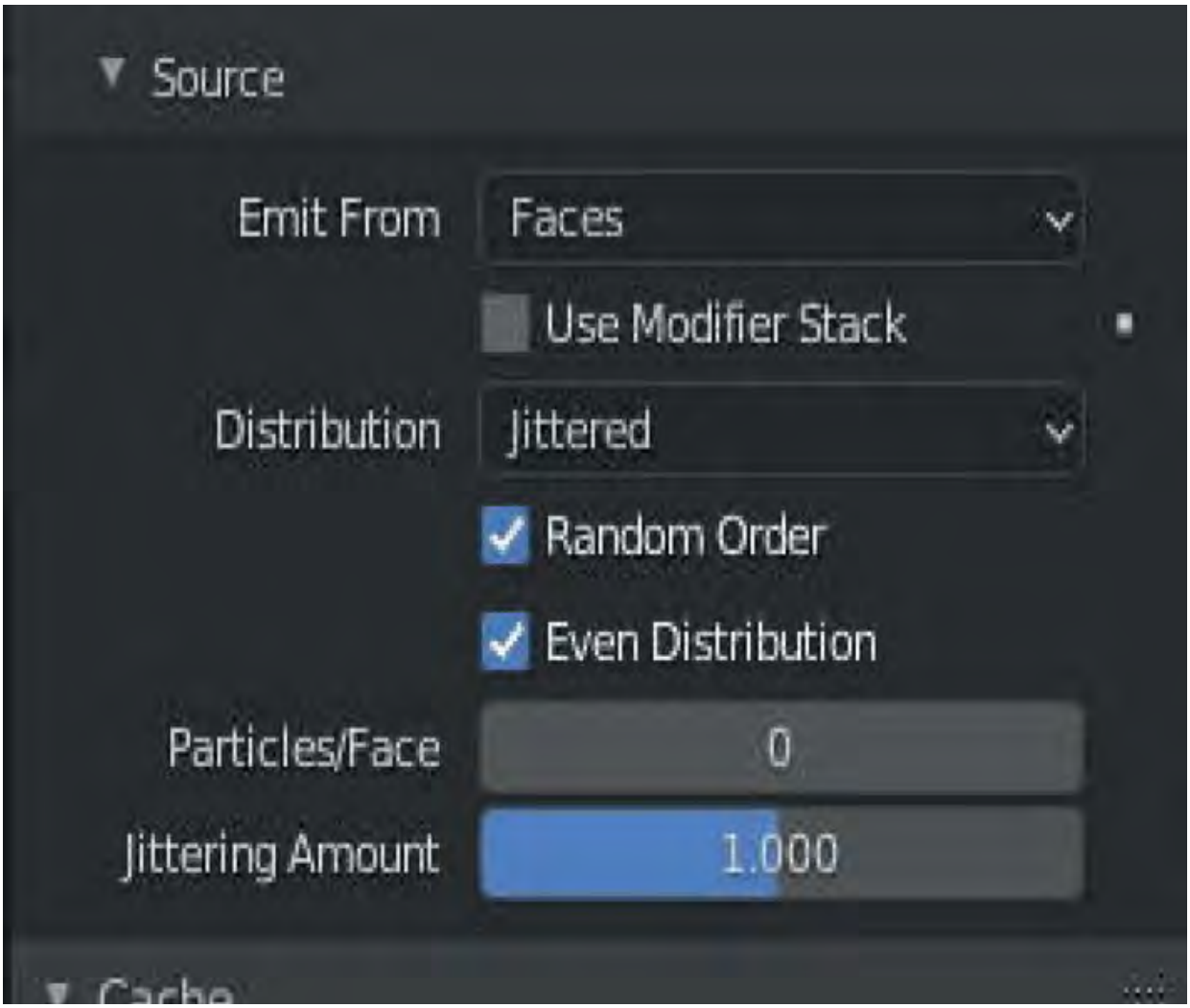


Figure 24.9

Emit From: Faces. The Particles Emit from the Faces of the Object's **Mesh**. The alternatives are from the **Vertices** or from **Volume** (the body of the mesh - See Section 24.7). The remainder of the settings in the Source Tab govern the order in which Particles are emitted (Figure 24.9).

24.4 The Cache Tab

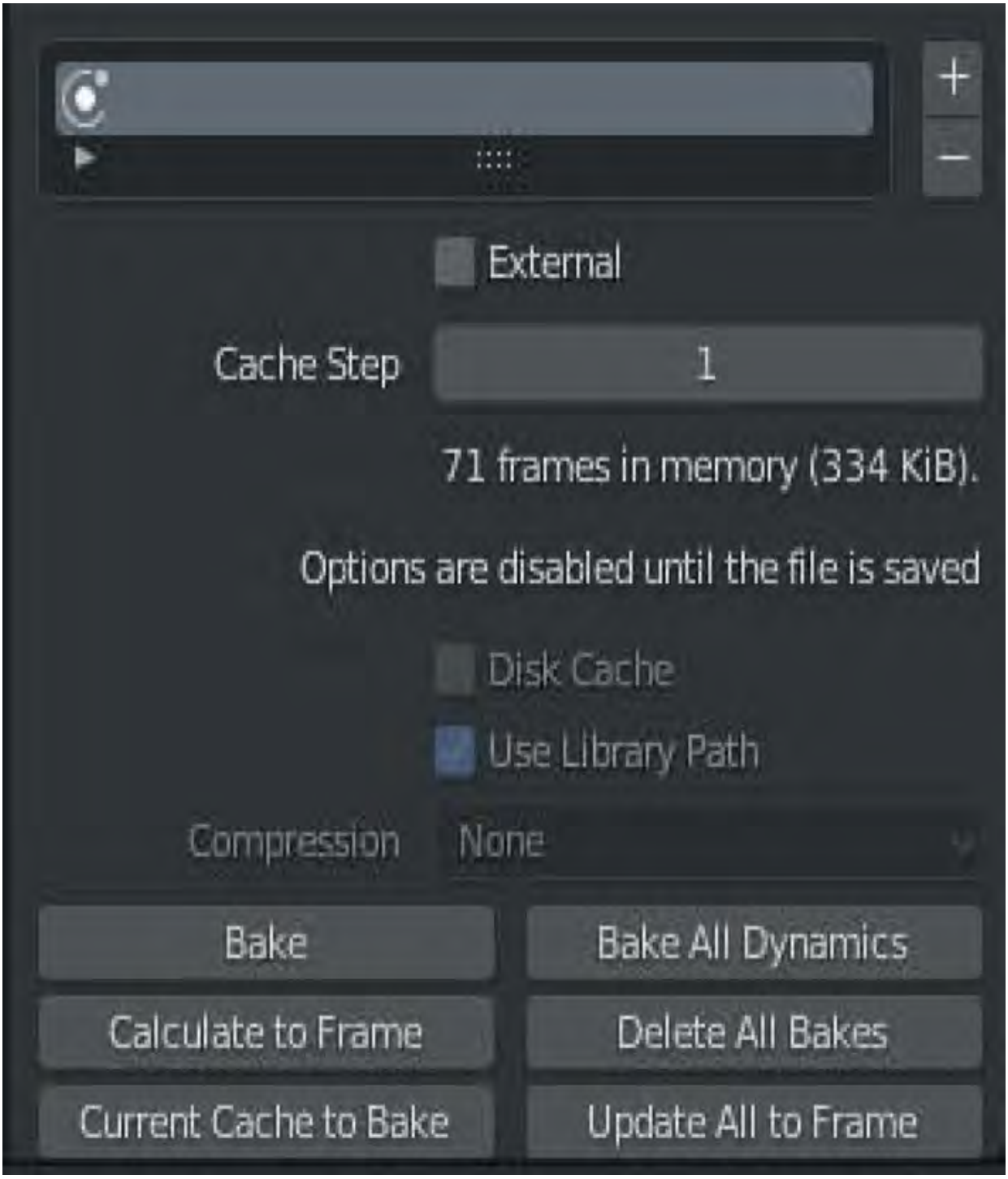


Figure 24.10

When a Particle System is played for the first time in the default Blender Scene the computer calculates the information required to display each Frame in the simulation (animation) and stores it in RAM (memory). When the simulation is played a second time the computer recalculates the information with any changes made to the settings and again stores the information in RAM. If the Blender file is closed without being saved the information is lost.

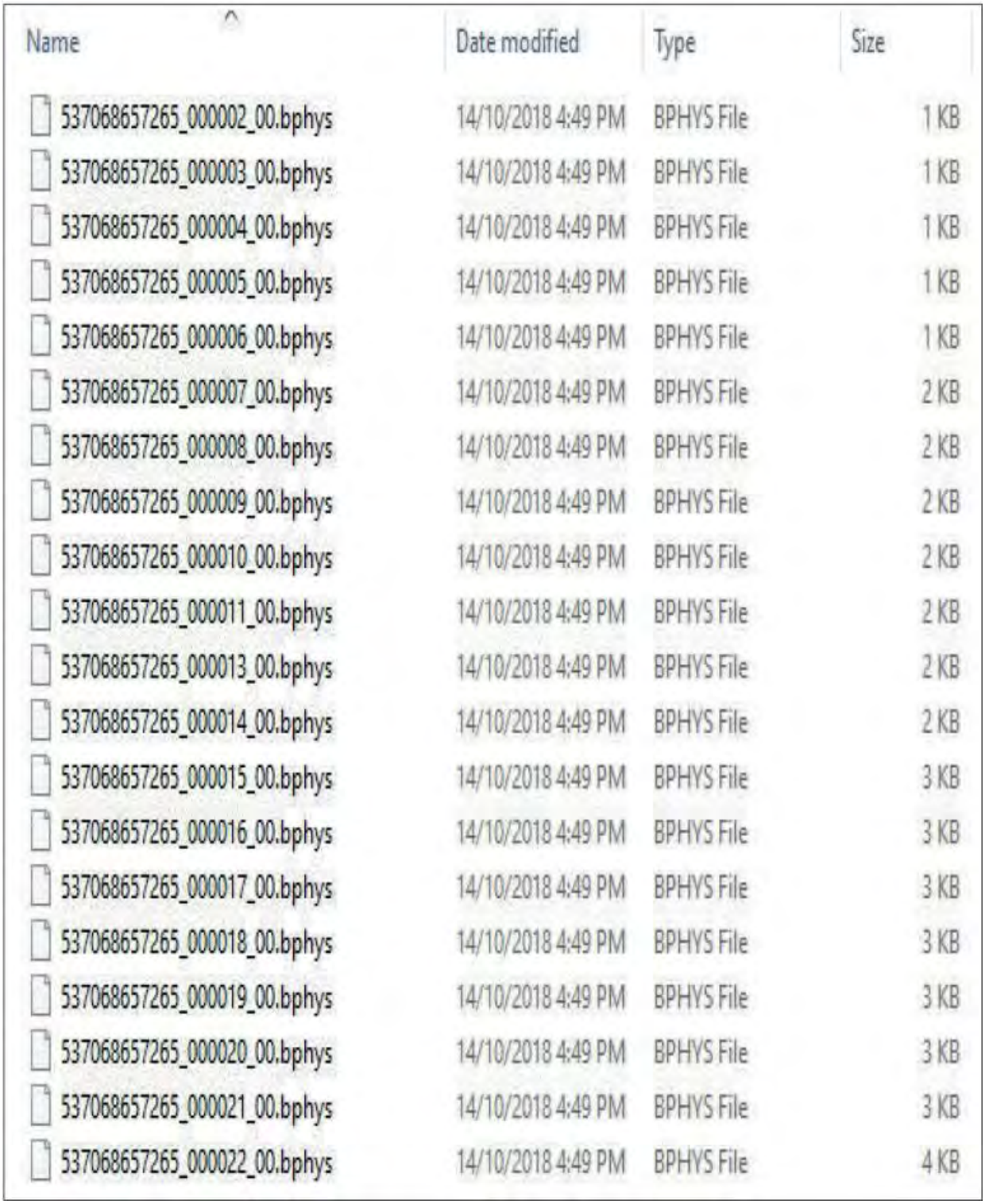
With a complicated Particle Simulation, writing data to RAM can use a considerable amount of memory which in turn can influence the performance of the computer. It is, therefore, advisable to save the Blender file as early as possible. With the file saved you have the option to save the simulation to a Cache which frees up memory.

With a complicated Particle Simulation, writing data to RAM can use a considerable amount of memory which in turn can influence the performance of the computer. It is, therefore, advisable to save the Blender file as early as possible. With the file saved you have the option to save the simulation to a Cache which frees up memory.

Note the statement in the **Cache Tab: Options are disabled until the file is saved**. This is basically saying, you can't save to the Cache until you have saved the Blender file. With the file saved check **Disk Cache** which will save the data to the Cache using the Library (Lib) Path.

Playing the simulation with **Disk Cache** checked creates a blend cache_Cache file and places it in the same directory as the .blend file. When Disk Cache is checked after saving the file you will see a red line at the bottom of the Timeline Editor indicating the data that has been saved. Playing the simulation with the default settings creates a solid line since data is recorded for each Frame of the default 250 Frames. In the where the Blender file is saved will find the blendcache_Cache folder containing 250 BPHYS files.

blendcache_Cache	14/10/2018 4:49 PM	File folder	
Cache.blend	14/10/2018 4:47 PM	Blender File	656 KB



Name	Date modified	Type	Size
537068657265_000002_00.bphys	14/10/2018 4:49 PM	BPHYS File	1 KB
537068657265_000003_00.bphys	14/10/2018 4:49 PM	BPHYS File	1 KB
537068657265_000004_00.bphys	14/10/2018 4:49 PM	BPHYS File	1 KB
537068657265_000005_00.bphys	14/10/2018 4:49 PM	BPHYS File	1 KB
537068657265_000006_00.bphys	14/10/2018 4:49 PM	BPHYS File	1 KB
537068657265_000007_00.bphys	14/10/2018 4:49 PM	BPHYS File	2 KB
537068657265_000008_00.bphys	14/10/2018 4:49 PM	BPHYS File	2 KB
537068657265_000009_00.bphys	14/10/2018 4:49 PM	BPHYS File	2 KB
537068657265_000010_00.bphys	14/10/2018 4:49 PM	BPHYS File	2 KB
537068657265_000011_00.bphys	14/10/2018 4:49 PM	BPHYS File	2 KB
537068657265_000013_00.bphys	14/10/2018 4:49 PM	BPHYS File	2 KB
537068657265_000014_00.bphys	14/10/2018 4:49 PM	BPHYS File	2 KB
537068657265_000015_00.bphys	14/10/2018 4:49 PM	BPHYS File	3 KB
537068657265_000016_00.bphys	14/10/2018 4:49 PM	BPHYS File	3 KB
537068657265_000017_00.bphys	14/10/2018 4:49 PM	BPHYS File	3 KB
537068657265_000018_00.bphys	14/10/2018 4:49 PM	BPHYS File	3 KB
537068657265_000019_00.bphys	14/10/2018 4:49 PM	BPHYS File	3 KB
537068657265_000020_00.bphys	14/10/2018 4:49 PM	BPHYS File	3 KB
537068657265_000021_00.bphys	14/10/2018 4:49 PM	BPHYS File	3 KB
537068657265_000022_00.bphys	14/10/2018 4:49 PM	BPHYS File	4 KB

Figure 24.11

250 BPHYS Data Files, one for each Frame in the Simulation

Figure 24.12

Timeline Editor – Red Line indicates files saved.

With the default simulation (250 Frames) there are 250 BPHYS Files. The longer the simulation the more files are created. To save space in the Cache, when you have a lengthy simulation, you may elect to only save data for some of the Frames. To do this increase the Cache Steps value. Increasing the value to 10 means every tenth frame is recorded, therefore, the number of BPHYS files in the Cache for the default 250 frame simulation would be 26 (25 divisions – Frame at each end). Increasing the Cache Steps saves space on the Hard Drive at the expense of a lesser quality in the simulation.

24.5 The Velocity Tab

The settings in the **Velocity Tab** control the direction and speed of the Particle Emission (Figure 24.13).

Figure 24.13

Normai: Gives the Particles an initial Velocity normal (at right angles to) the point of origin.

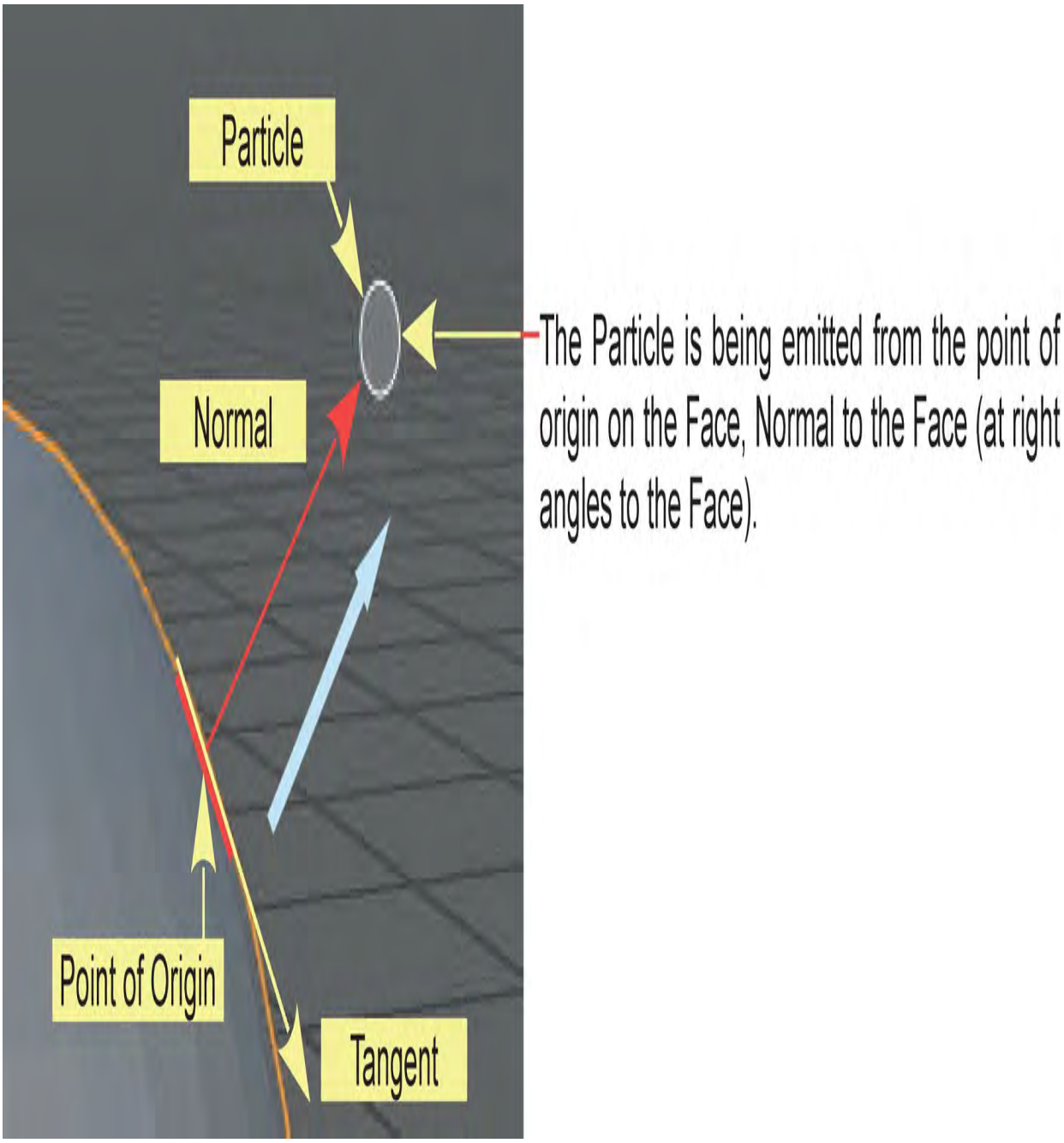
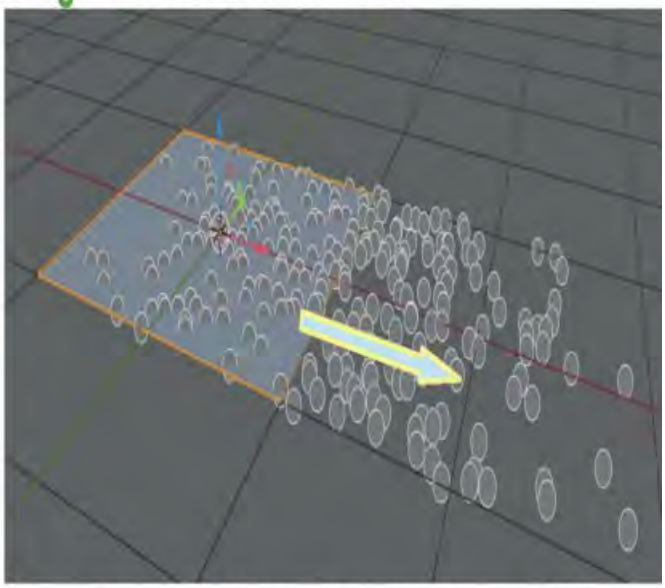


Figure 24.14

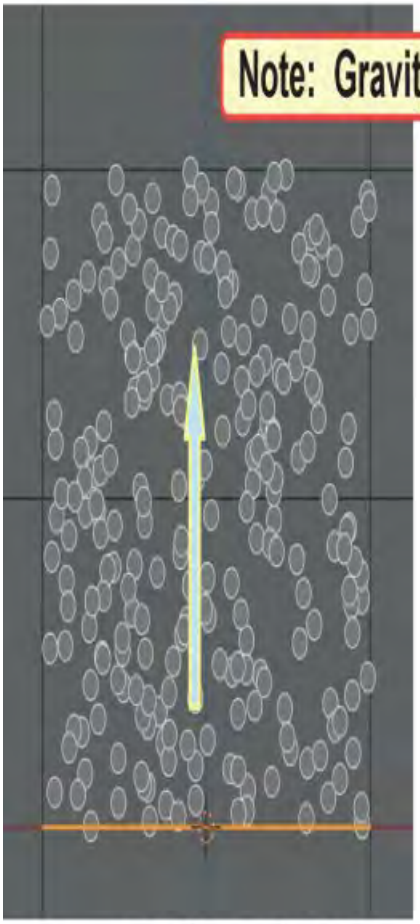
Tangent: Parallel to the Face.

Tangent: Parallel to the Face.

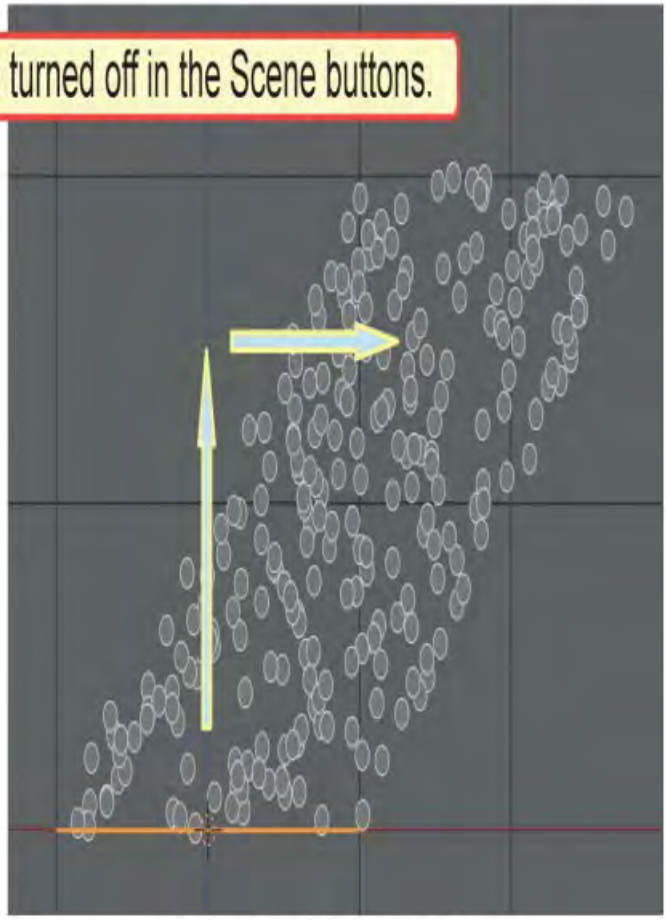


Tangent: 1m/s – Tangent Phase +0.5

Note: Gravity has been turned off in the Scene buttons.



Normal: 1.000 – X: 0.000



Normal: 1.000 – X: 1.000

Figure 24.15

Tangent Phase and Object Alignment: Both Control the Emission direction between Normal and Tangent.

Understanding Tangent Phase

To assist in understanding **Tangent Phase** study the following diagrams (Figure 24.16 - 17)

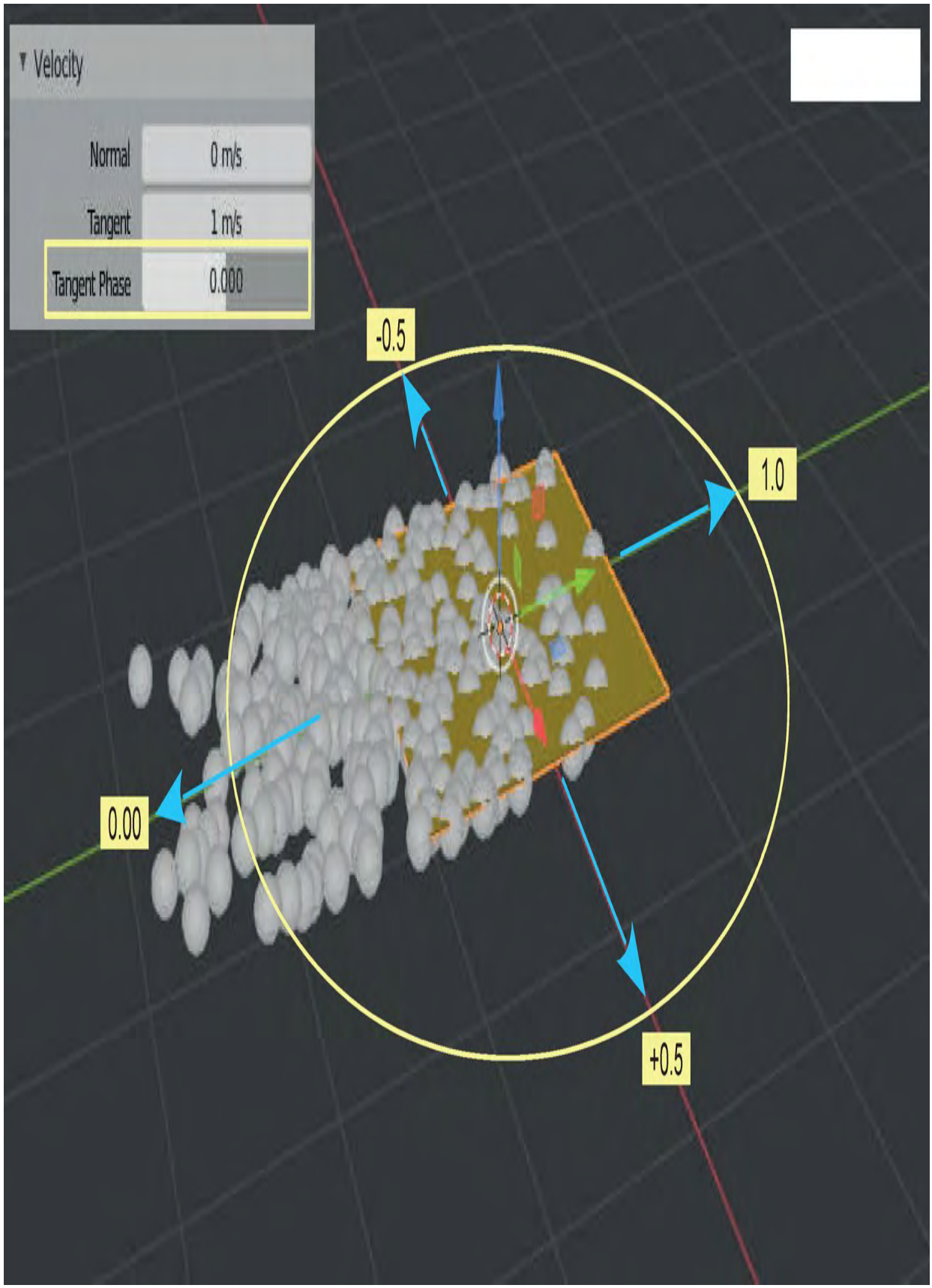

Velocity
Normal
0 m/s
Tangent
1 m/s
Tangent Phase
0.000
-0.5
1.0
0.00
+0.5

Figure 24.16

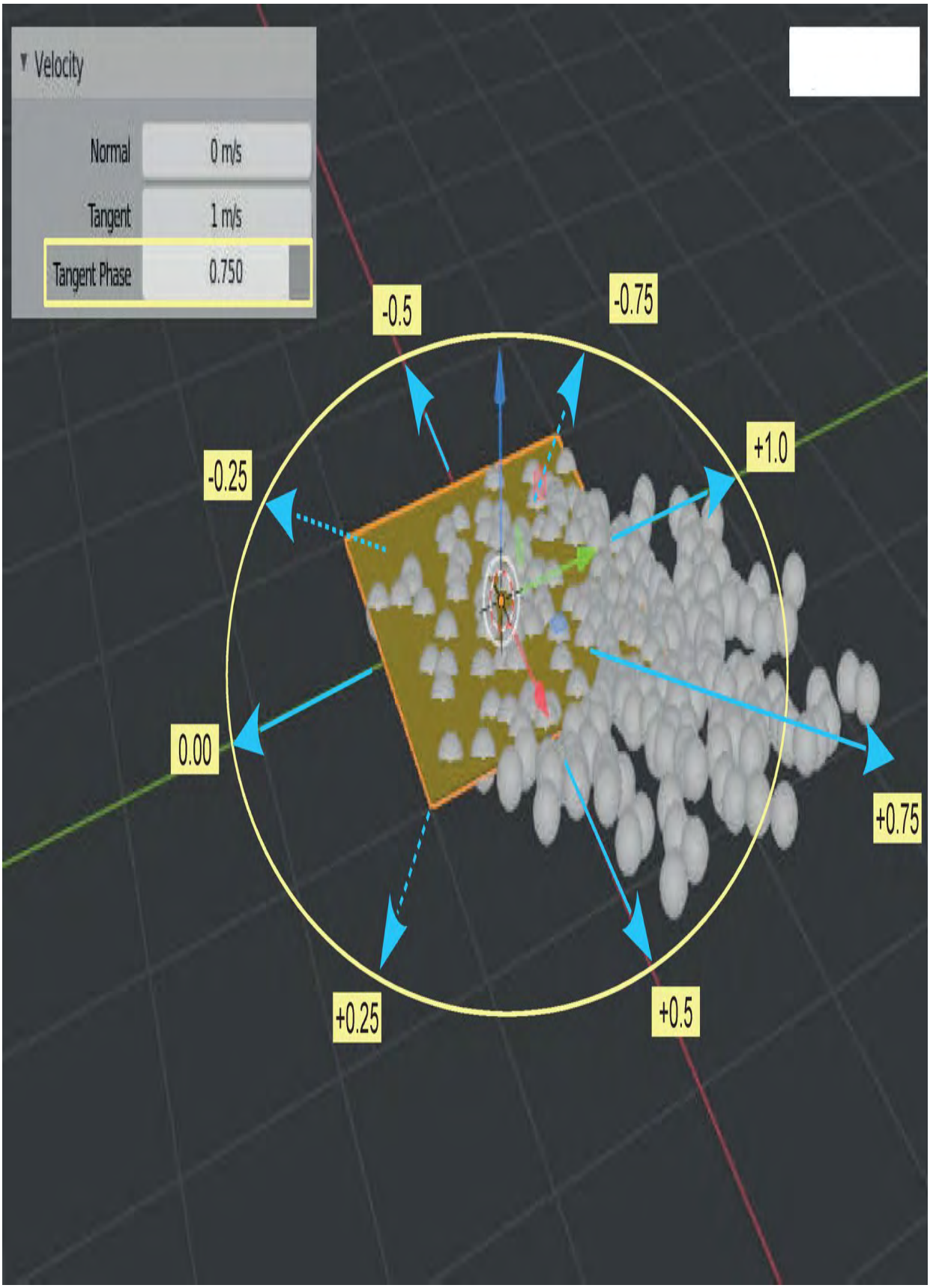

Velocity
Normal
0 m/s
Tangent
1 m/s
Tangent Phase
0.750
-0.5
-0.75
-0.25
+1.0
0.00
+0.75
+0.25
+0.5

Figure 24.17

By adjusting the **Tangent Phase** values in the Properties Editor. Particle Properties the direction of Particle Emission is controlled. Pay attention to positive and negative value.

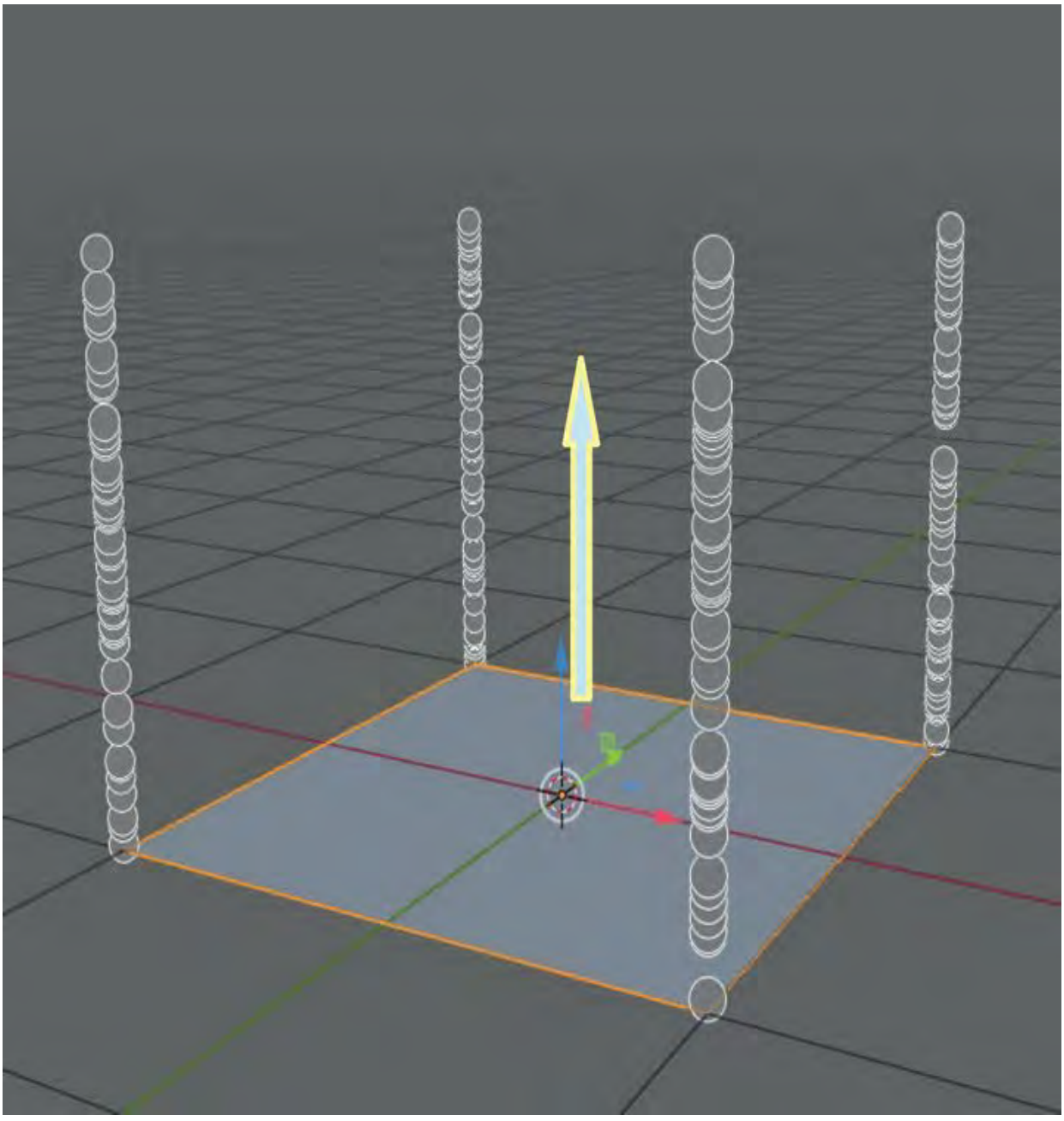


Figure 24.18

Figure 24.14 shows Particles being emitted from the single Face of the Plane Object. By changing Face to **Vertices** in the **Source Tab,** with the default values (Normal: 1.000 – X: 0.000) the Particles are emitted from the four **Vertices** of the Plane, normal to the face of the Plane (Figure 24.17).

At this point you have just enough information to control the Emission of Particles but they are just Particles, little white spheres, and not particularly interesting. The spheres represent positions on the computer Screen for the display of other Objects with the objective being, to create visual displays.

24.6 Particle Display

A Particle will display as an Object which has been added to a Scene.

To demonstrate; have a Plain Object and a Monkey Object in a Scene. Scale the Monkey way down and park it off to one side (Figure 24.18).

Have the Plane selected in the 3D Viewport Editor and apply a Particle System. **Turn off Gravity in the Scene buttons**.

In the **Emission Tab** (Figure 24.19) in the Particles buttons for the Plane decrease the Number to 5 (Emit 5 Particles only) and change the End value to 10 (the 5 Particles will be Emitted in 10 Frames). In the **Render Tab** (Figure 24.20) change **Render**

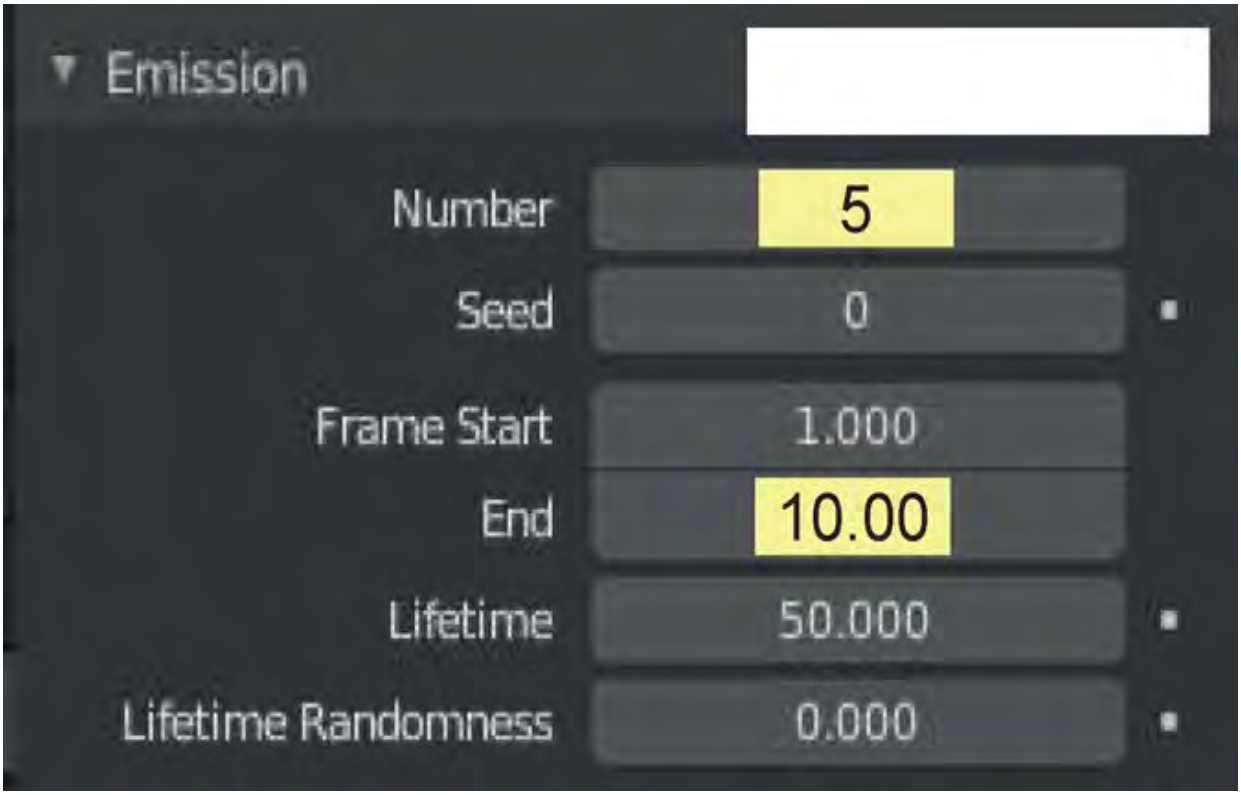

Emission
Number
5
Seed
0
Frame Start
1.000
End
10.00
Lifetime
50.000
Lifetime Randomness
0.000

Figure 24.19

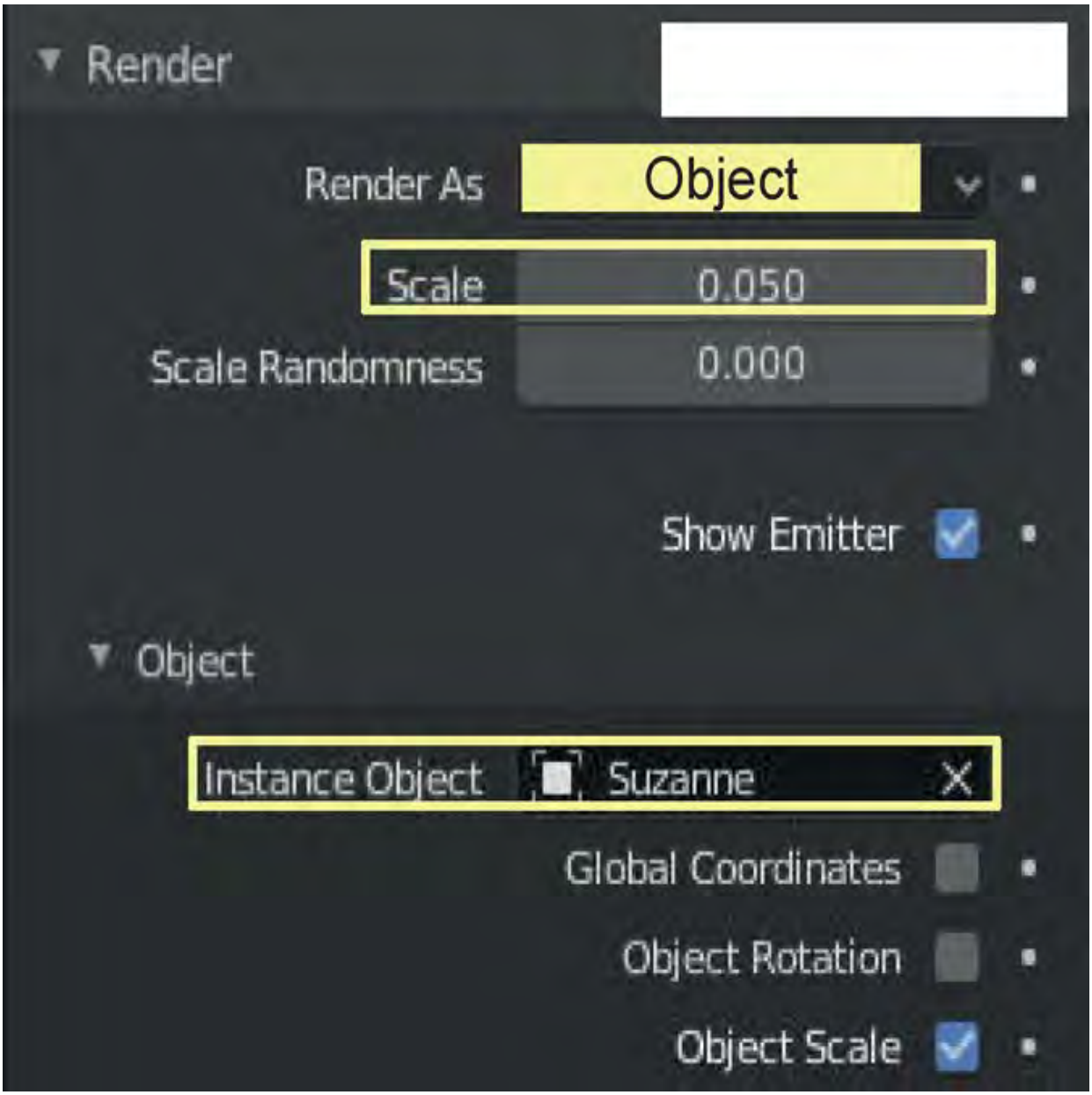


Figure 24.20

As to Object. Selecting Object introduces an Object Tab. Click where you see **Instance Object** and select **Suzanne** (Monkey).

Play the animation in the Timeline Editor (you may stop at Frame 50 since the Particles will only display for 50 Frames). Position the Timeline Cursor at Frame 45 then zoom in on the Plane. If you look closely you will see five tiny Monkeys sitting above the Plane. In the **Render Tab** increase the **Scale**.

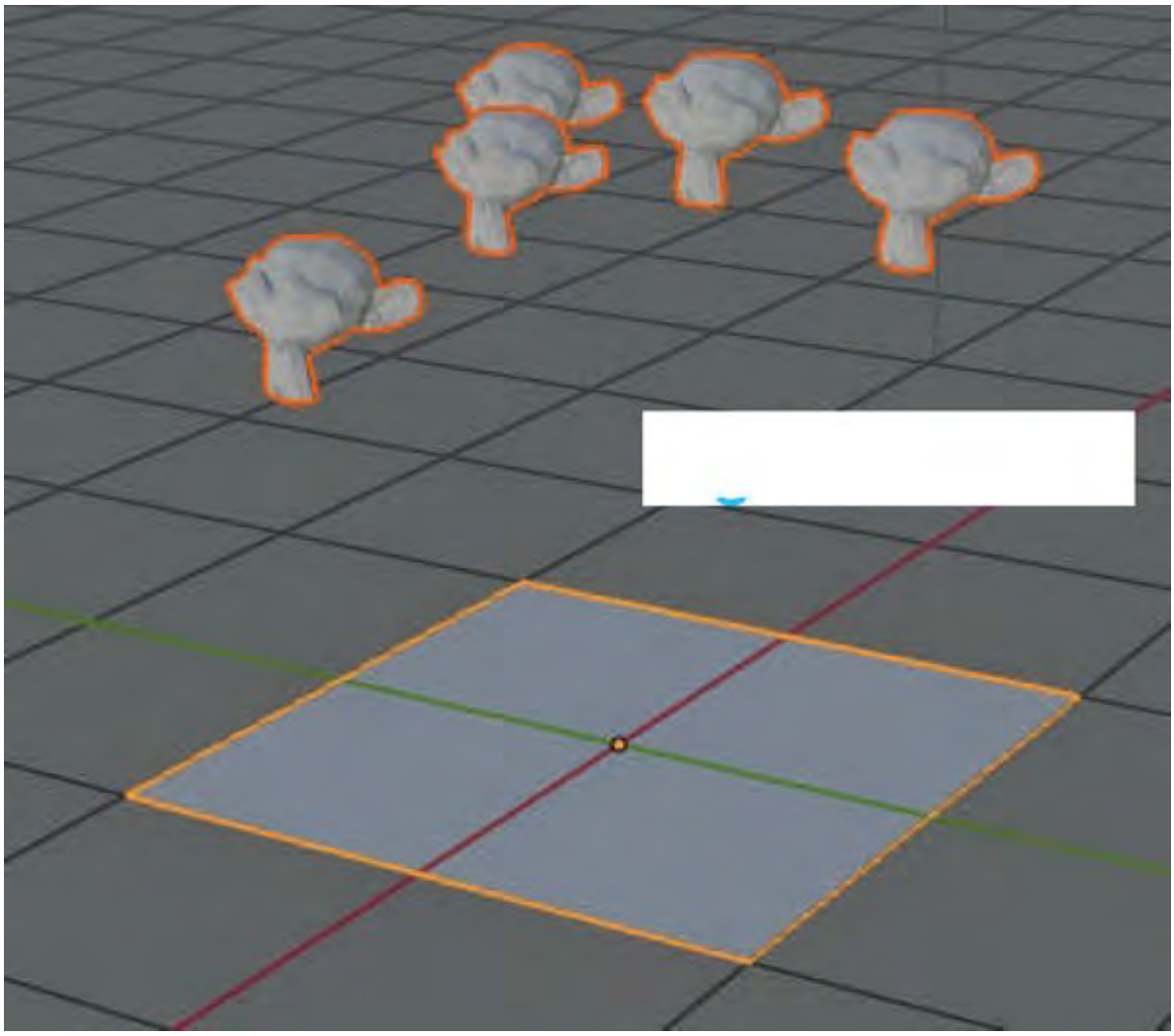


Figure 24.21

24.7 Particle Emission Options

The options for Particle Emission have been briefly mentioned when discussing the Source Tab in 24.3 and Emission from Vertices demonstrated in 24.5, Figure24.17. To clarify the options look at the default Cube Object in the default 3D Viewport Editor. Have the Cube displayed in **Wireframe Display Mode**. Click the button in the Header (upper RH side of Screen) (Figure 24.22).

The Particle Emission options are accessed in the Properties Editor, Particle buttons, Source Tab (Figure 24.23).

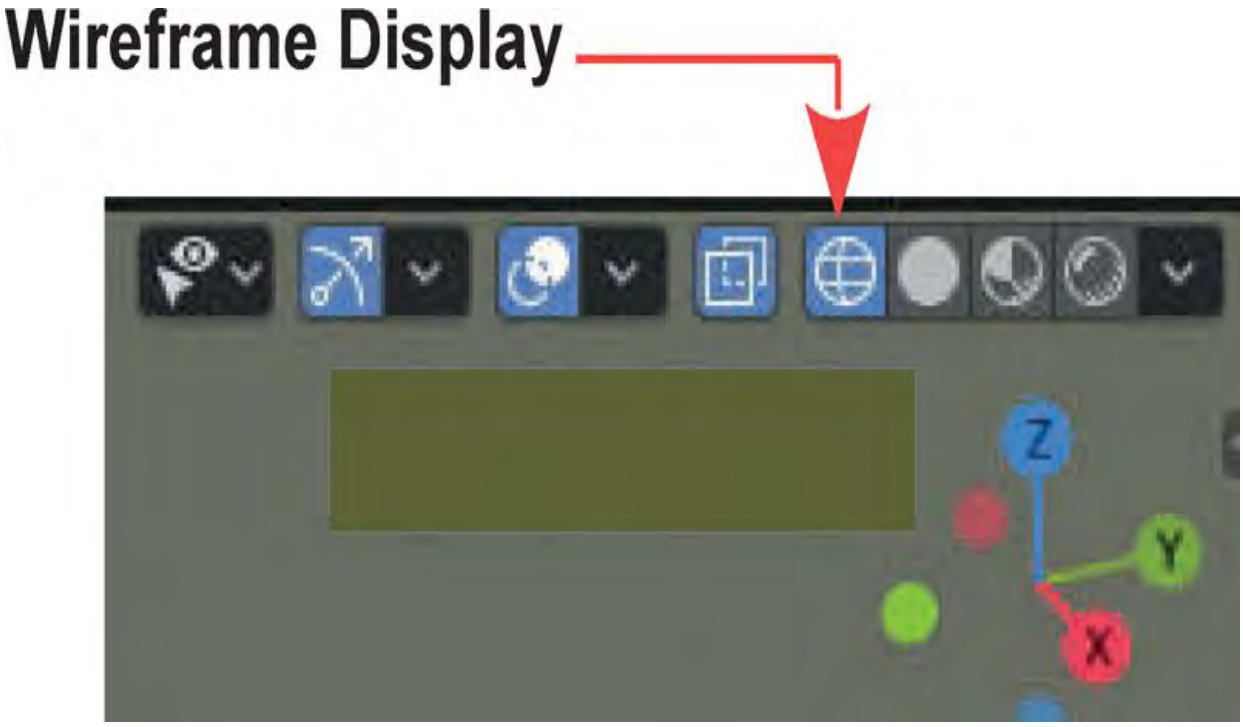


Figure 24.22

Figure 24.23

Figure 24.24 shows the default Cube with the default Particle System applied at Frame 1 (One Particle Showing). The default **Emit From** value in the **Source Tab** is: **Faces** (Figure 24.23).

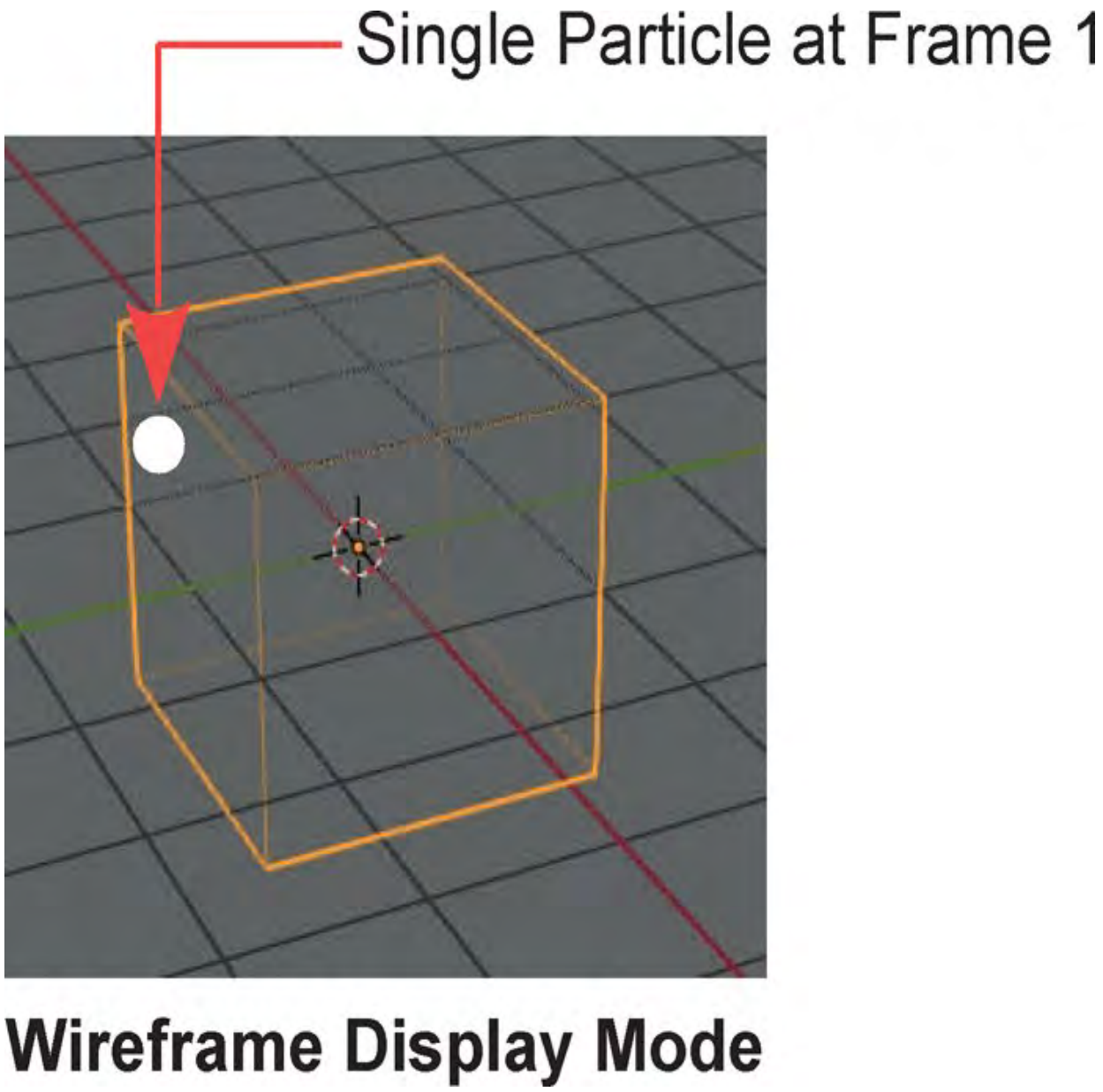
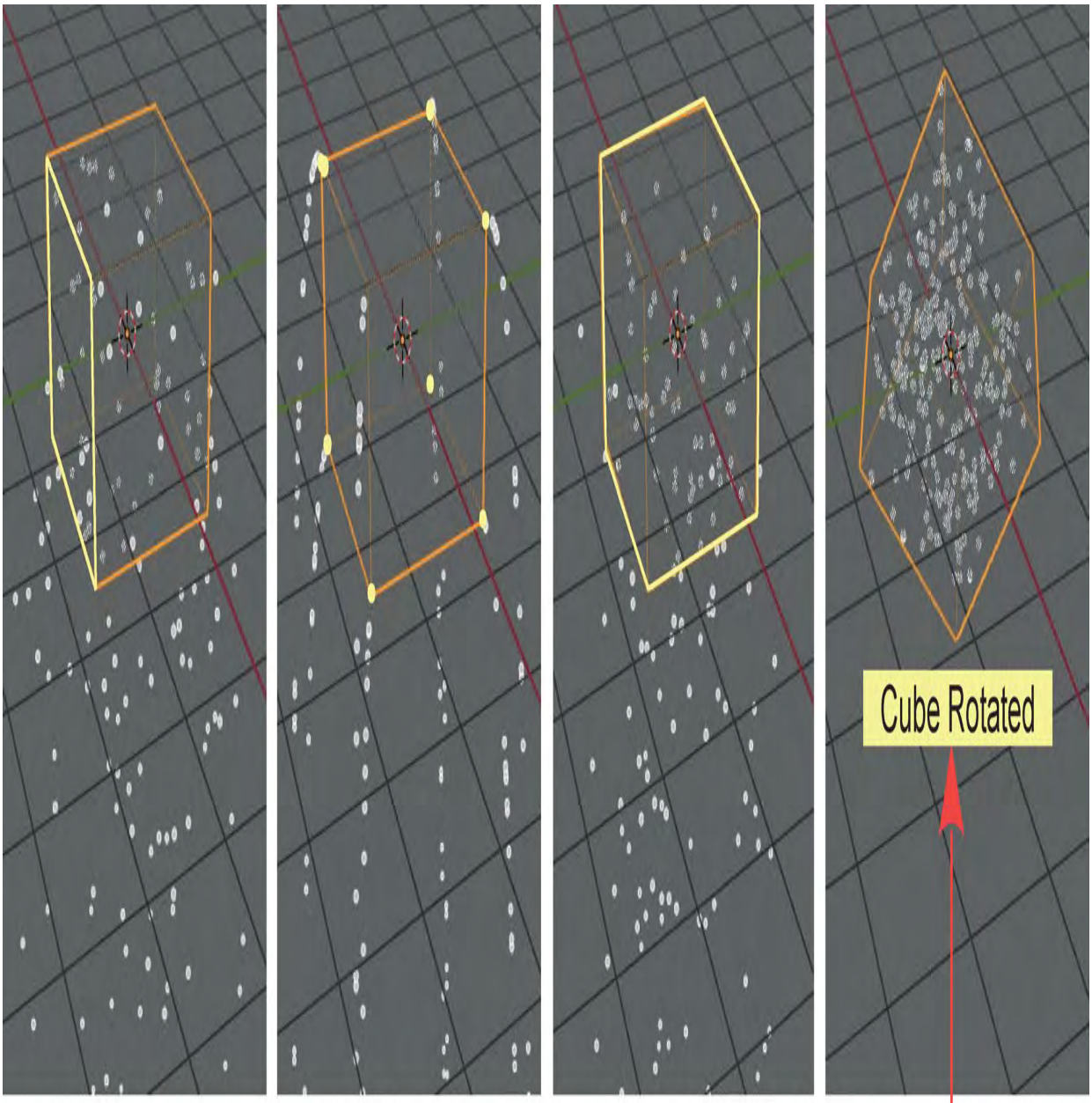


Figure 24.24

Wireframe Display Mode

Figure 24.25 shows particles being Emitted with the different options selected.



Emit From: **Faces**

Emit From: **Verts**

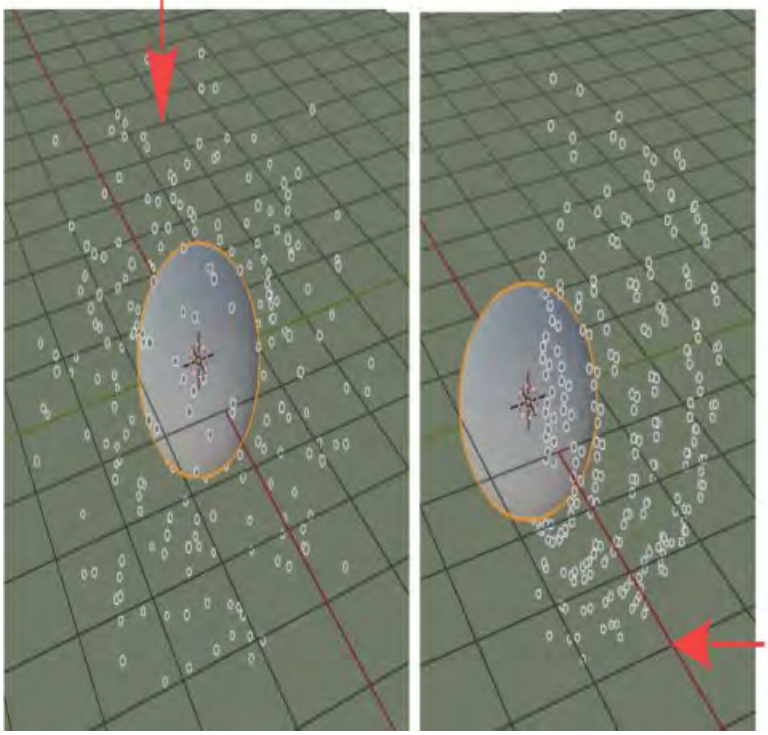
Emit From: **Volume**

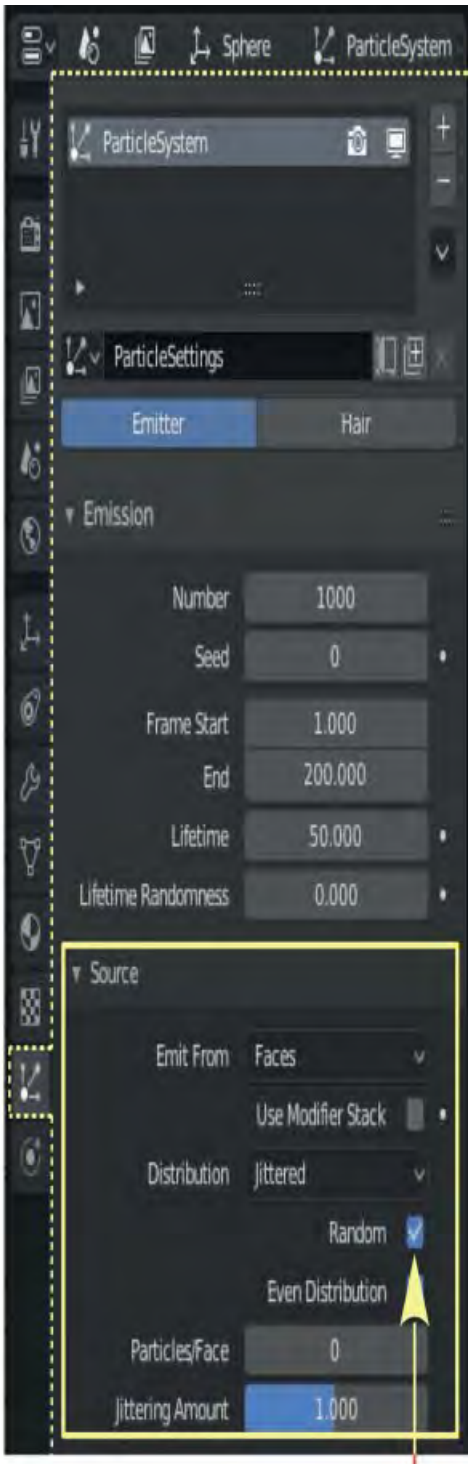
Emit From: Volume with Velocity Normal = 0.00 and Gravity turned off. The Particles accumulate inside the Volume of the Cube.

Figure 24.25

24.8 Order of Emission

By default Particles are set to **Emit From: Faces** in a **Random Order** (Distribution).





In the Source Tab **uncheck Random** and replay the animation.

Figure 24.26

To demonstrate the Order of Emission options, replace the default Cube in the 3D Viewport Editor with a UV Sphere. The Sphere has significantly greater number of Vertices and Faces from which to Emit Particles.

Disable Gravity in the Scene buttons.

The Order Of Particle Emission is controlled in the Properties Editor, Particle buttons, **Emission Source Tab** (the Source Tab only displays when the Emission Tab is opened).

The 3D Viewport Editor, by default, is in **User Perspective View,** therefore, it is difficult to see what has been achieved by removing the Random tick. Change the view to **Top Orthographic View** (Figure 24.28) then to **Front Orthographic View** (Figure 24.29). With the Timeline Editor Cursor, advanced to Frame 50, you will see an ordered array of Particles.

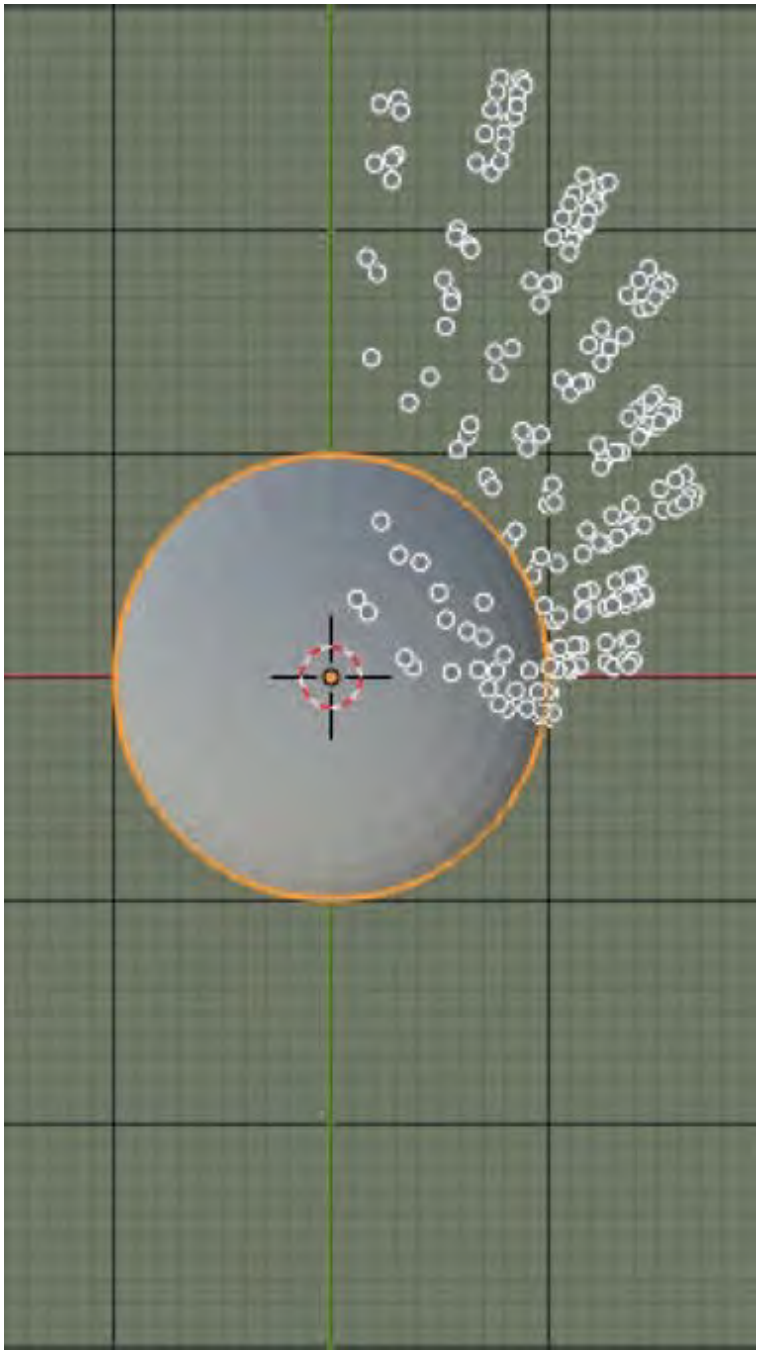


Figure 24.28

In both views the Particles are being Emitted from Faces.

By changing Emit From to **Verts** in the Source Tab (figure 24.30) the array of Particles is even more ordered (Figure 24.31 over).

Figure 24.29

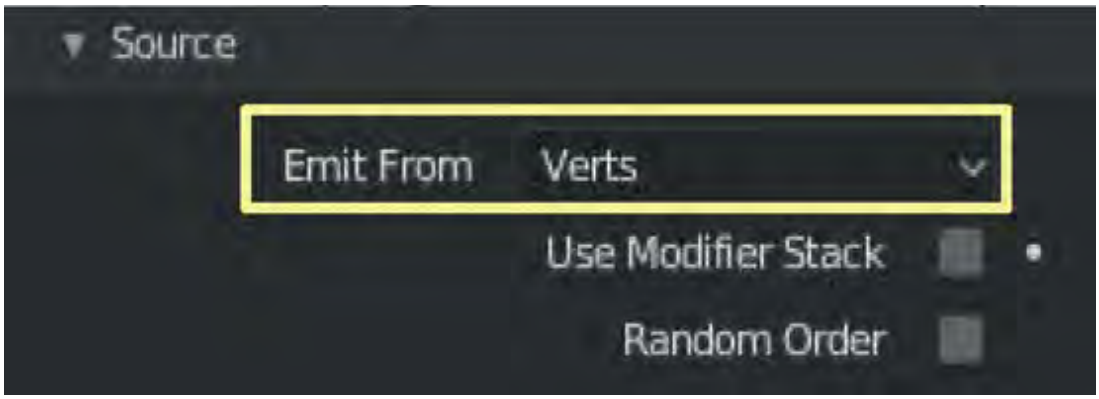


Figure 24.30

Particles Emitted from Vertices at Frame 50

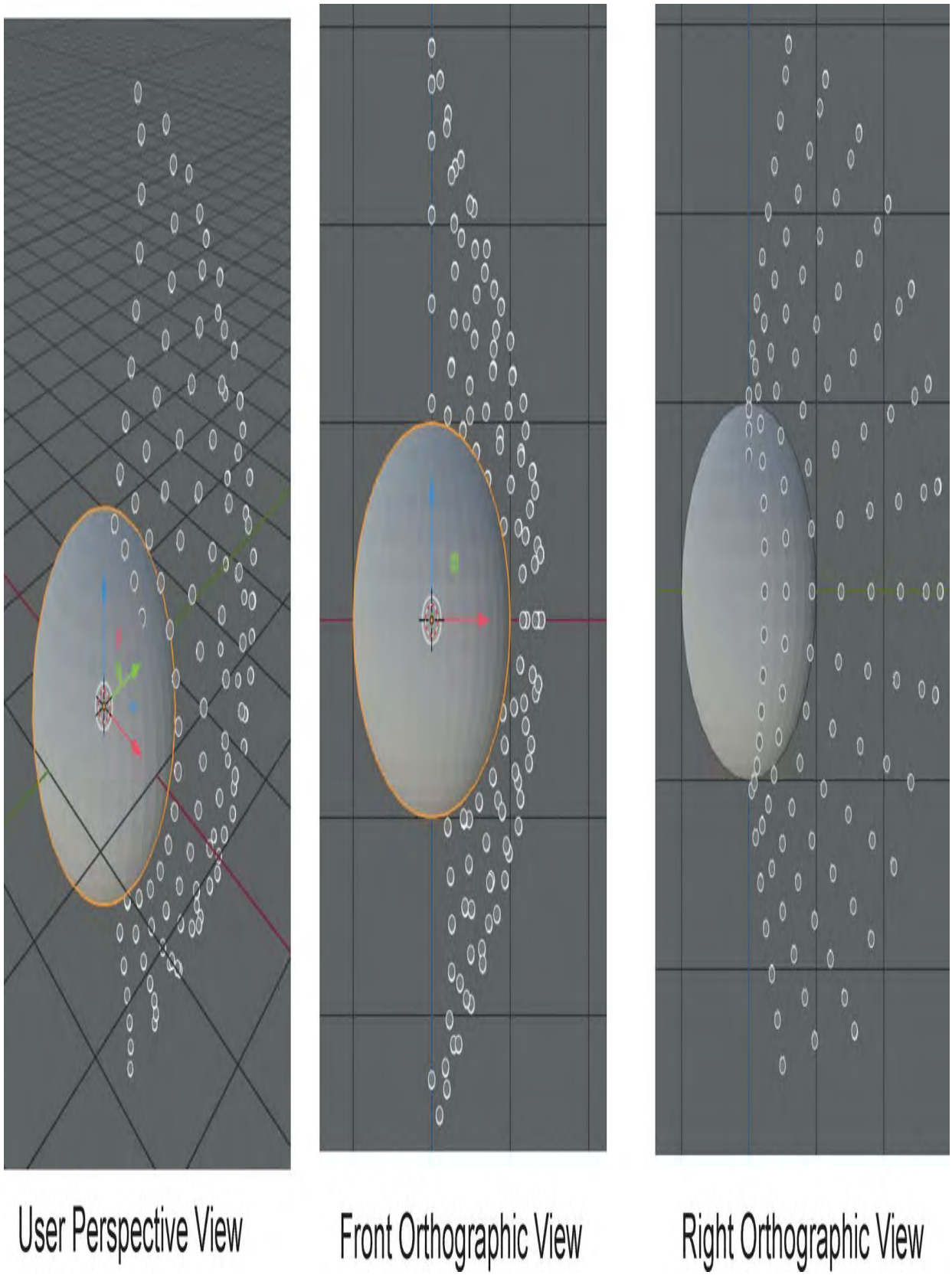
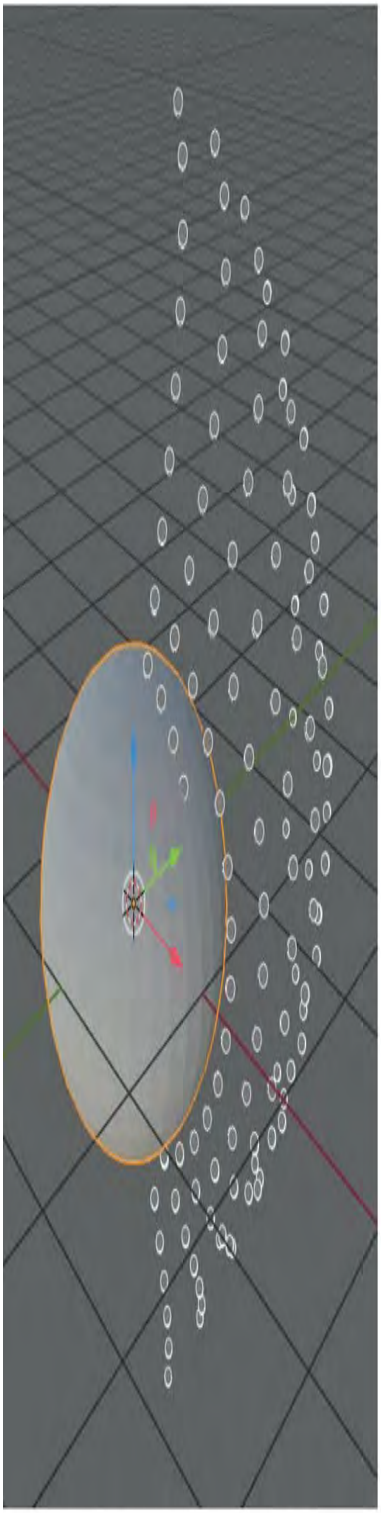
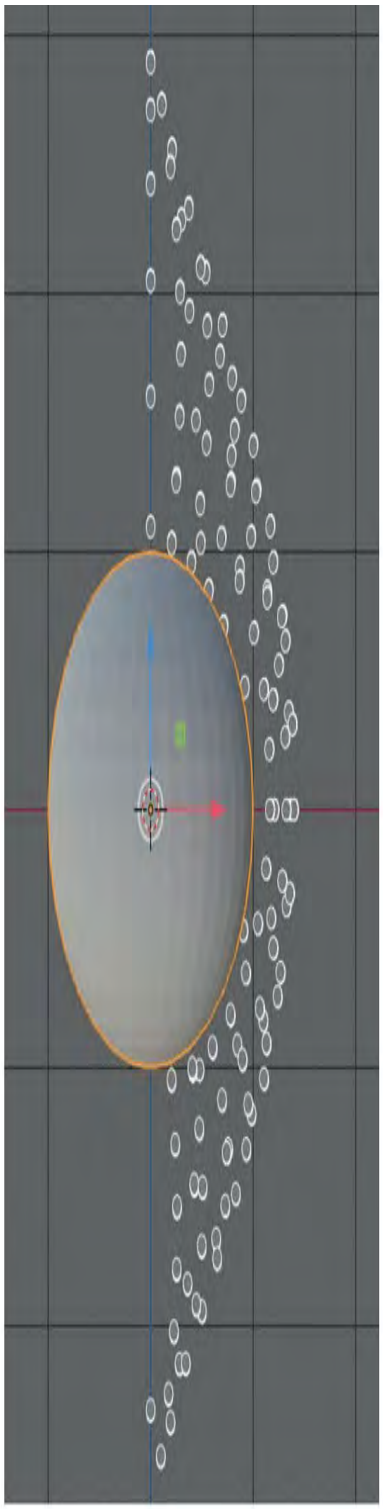

User Perspective View
Front Orthographic View
Right Orthographic View

Figure 24.31

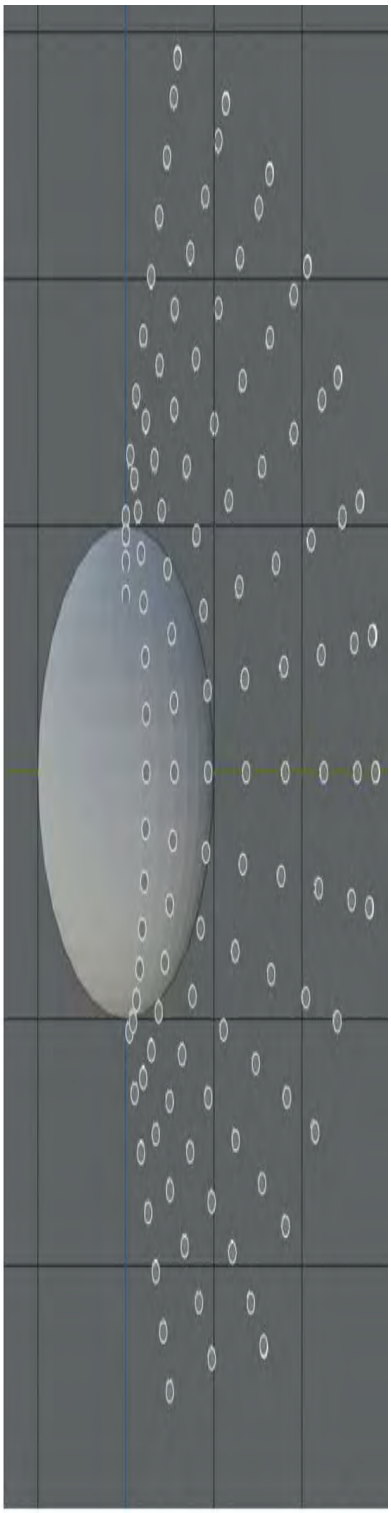
Particles Emitted from Faces at Frame 50



User Perspective View



Front Orthographic View



Right Orthographic View

24.9 Normals

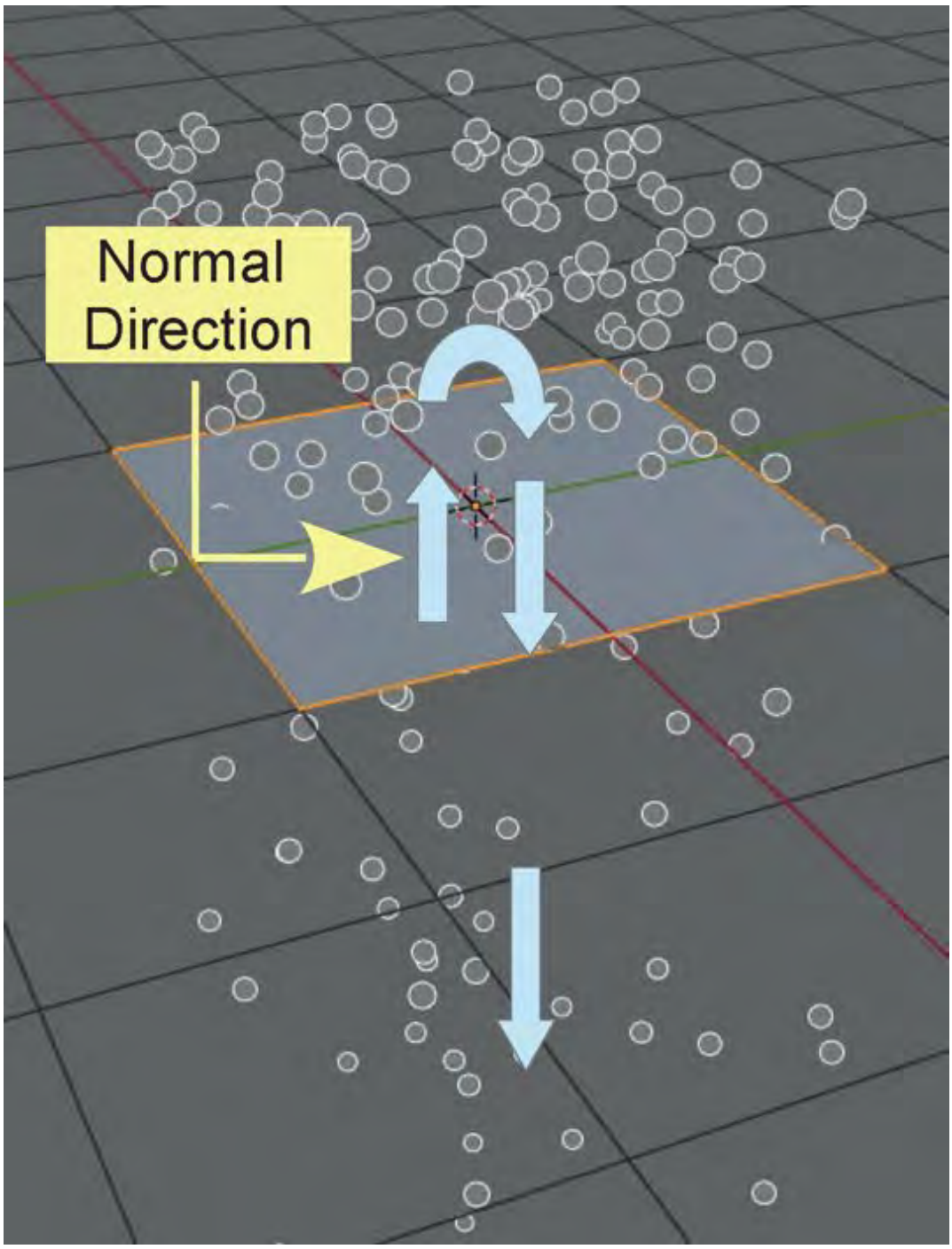


Figure 24.32

Particle Effects may be created by using different shaped Objects as Particle Emitters and manipulating **Normal** values in the Particle buttons (see 24.5 The Velocity Tab – Normals). Understanding how to control Normals is a key factor.

Have a Plane Object in the 3D Viewport Editor and apply a Particle System. When the animation is played in the Timeline Editor, Particles are emitted from the Face of the Plane and descend in the Scene due to the Gravitational force. In the **Velocity Tab** increase the Normal value to 5.000 and replay the animation.

You will observe that the Particles rise from the surface of the Plane before descending (Figure 24.32).

In **Object Mode** click on **Overlays** in the Header (Figure 24.33) and check **Face Orientation**. In the default Scene the upper

Figure 24.33

The Particles are emitted from the surface, Normal to the Face. In the default Particle System the direction of the Normal is upwards. Rotate the Plane 45° about the X Axis (R Key + X Key +45).

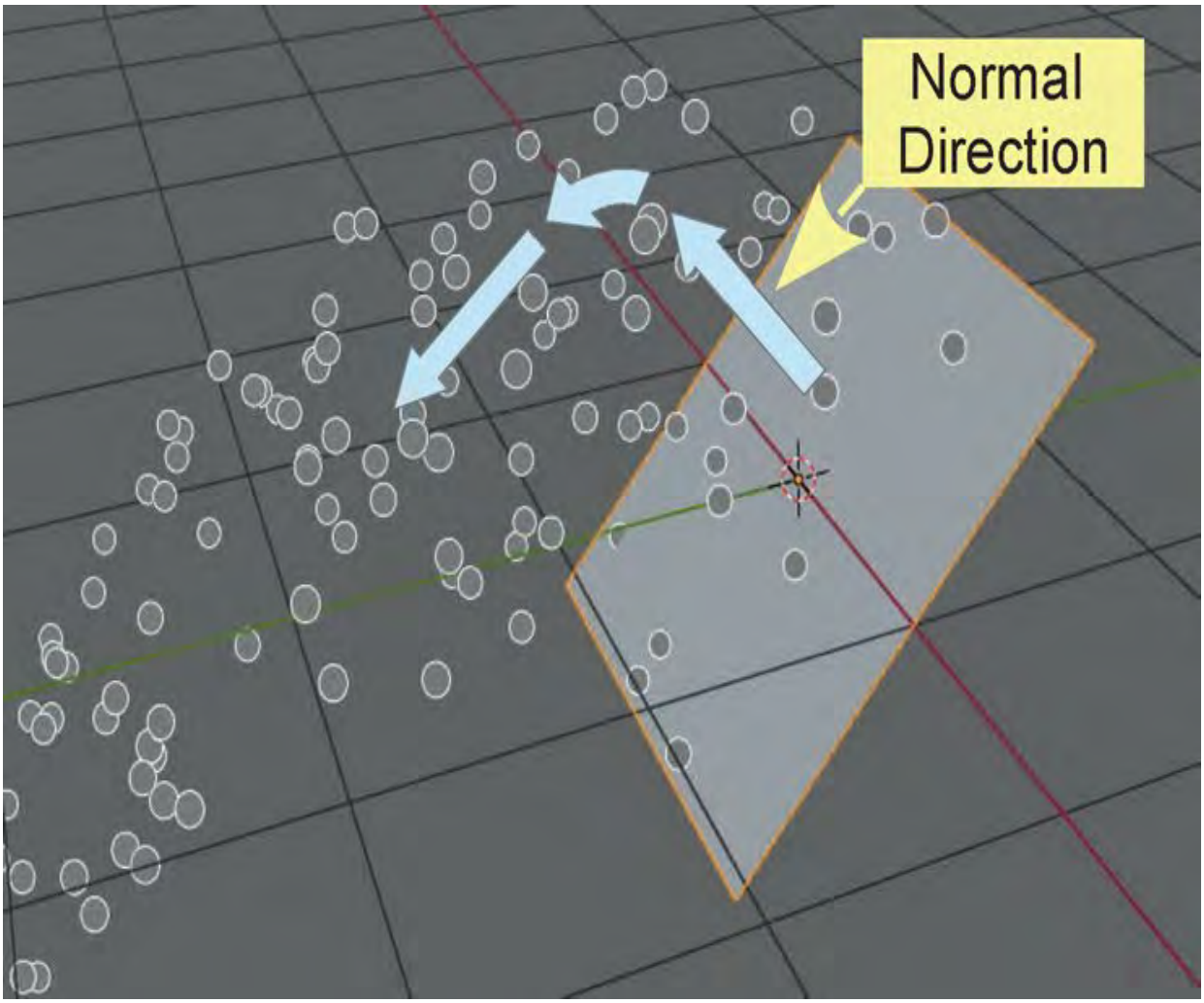


Figure 24.34

With the Plane rotated Particles continue to be emitted Normal to the Face of the Plane (Figure 24.34). The direction of the Normal is relative to the Face.

The direction of Normals may be visualised in two ways.

Motion Tracking In **Object Mode** click on **Overlays** in the Header (Figure 24.33) and check **Face Orientation**. In the default Scene the upper surface of the Plane displays blue indicating the positive direction for Emission. By rotating the view you will see the underside of the Plane is displayed red (negative) (Figure 24.35). Bear in mind that you can enter positive and negative values in the Velocity Tab which change the direction of Emission. This does not change the color display on the surface.

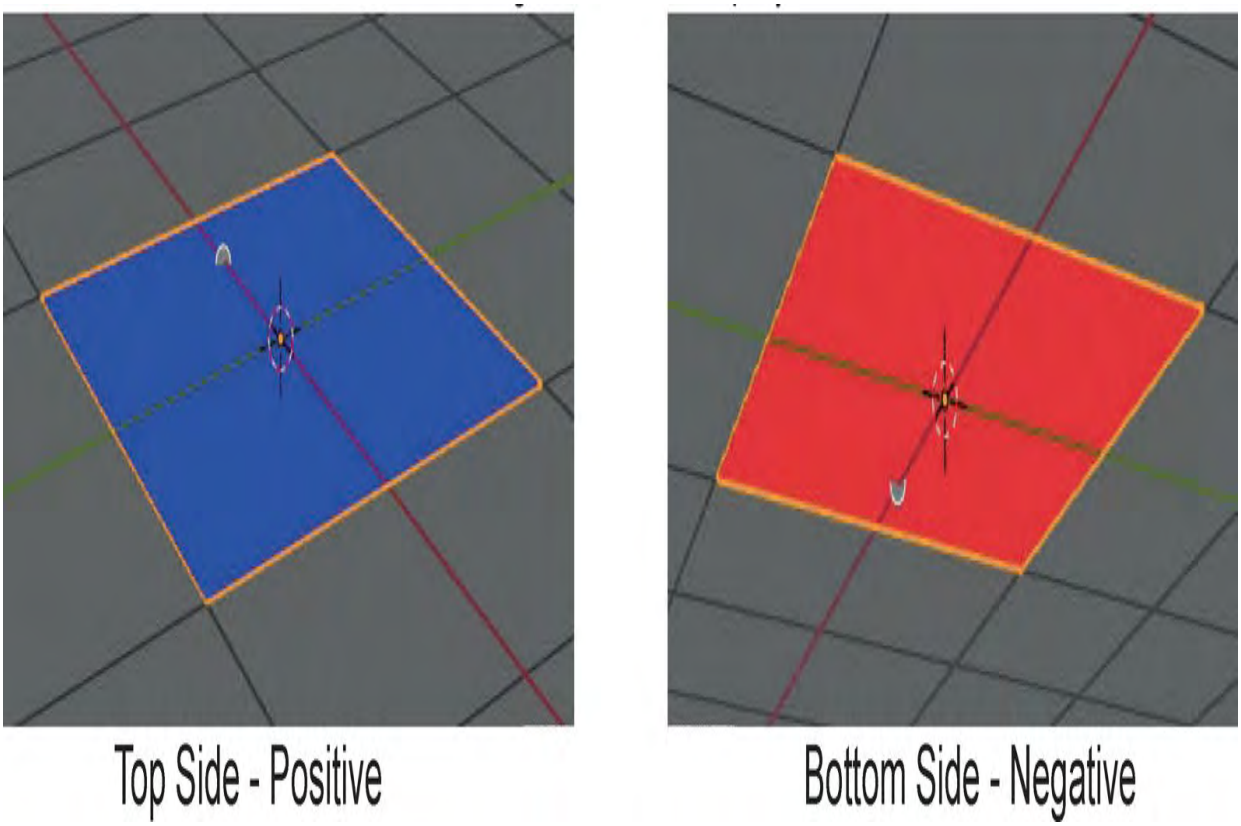


Figure 24.35

In **Edit Mode Overlays** (at the bottom of the panel) you will find **Normals** (Figure 24.36).

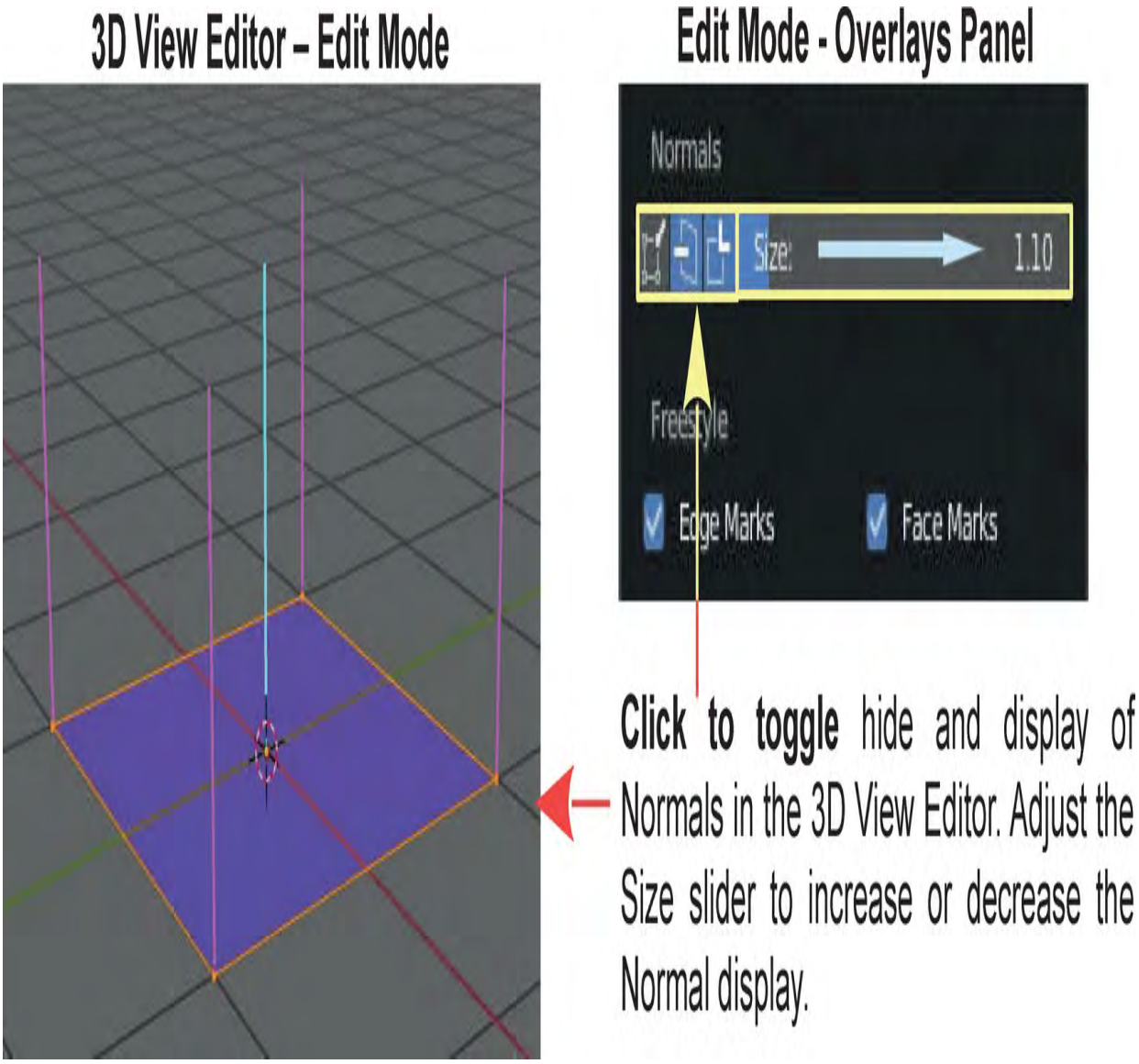


Figure 24.36

Note: Adjusting the Size value does not change the Velocity for emission.

Knowing how Particles will be Emitted from an Object allows you to set up a Particle Display.

As an example, construct a flat disk Object as shown in Figure 22.35 by selecting a Circle Object in Edit Mode. Press the E Key (Extrude – DO NOT

Move the Mouse). The Vertices are duplicated. Scale the duplicated Vertices in.

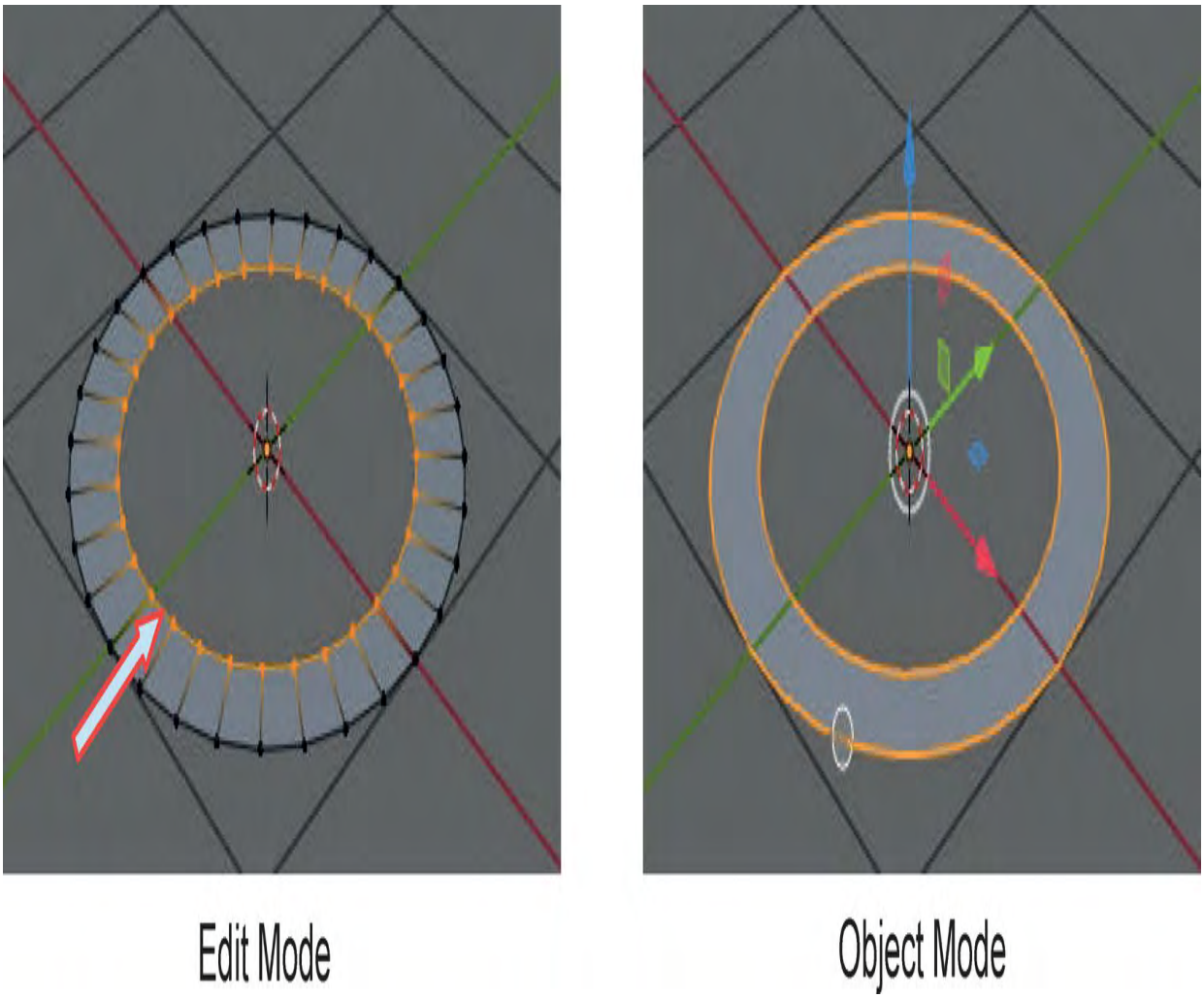


Figure 24.37

Turn off **Gravity** in the **Properties Editor, Scene buttons, Gravity tab**.

With the Disk Object selected in the 3D Viewport Editor, in Object Mode, add a **Particle System** leaving the default values in place. Play the animation in the Timeline Editor.

Particles will be Emitted from the Faces of the Disk and **fall** towards the bottom of the Screen despite Gravity being turned off. (Figure 24.38).

Particles are Emitted with a default starting velocity of **Normal = 1**. The velocity value is seen in the **Particle Properties, Velocity Tab**.

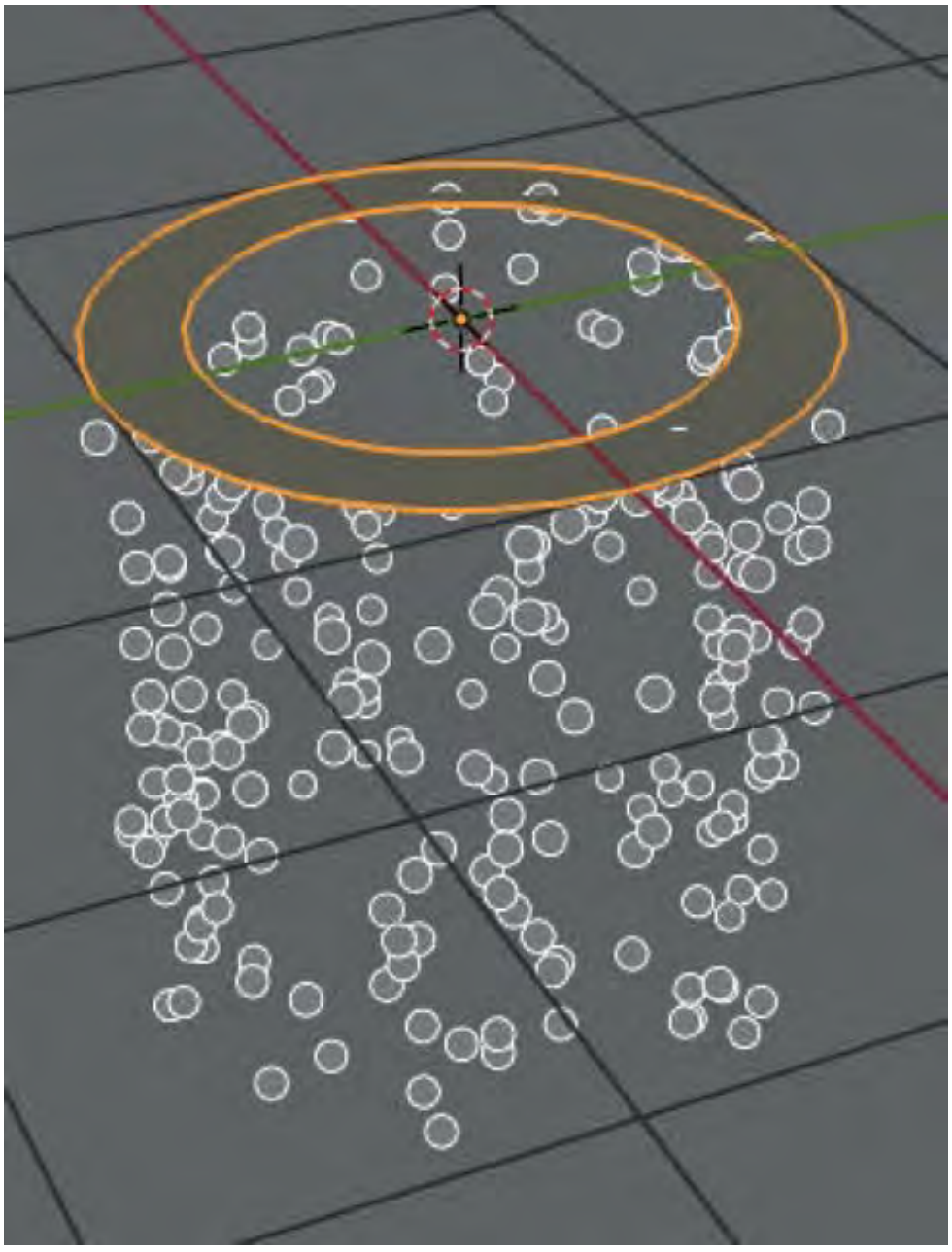


Figure 24.38

By turning on Normal visualisation as previously described you can see that the Normal direction is down, hence the descent (Figure 24.40).

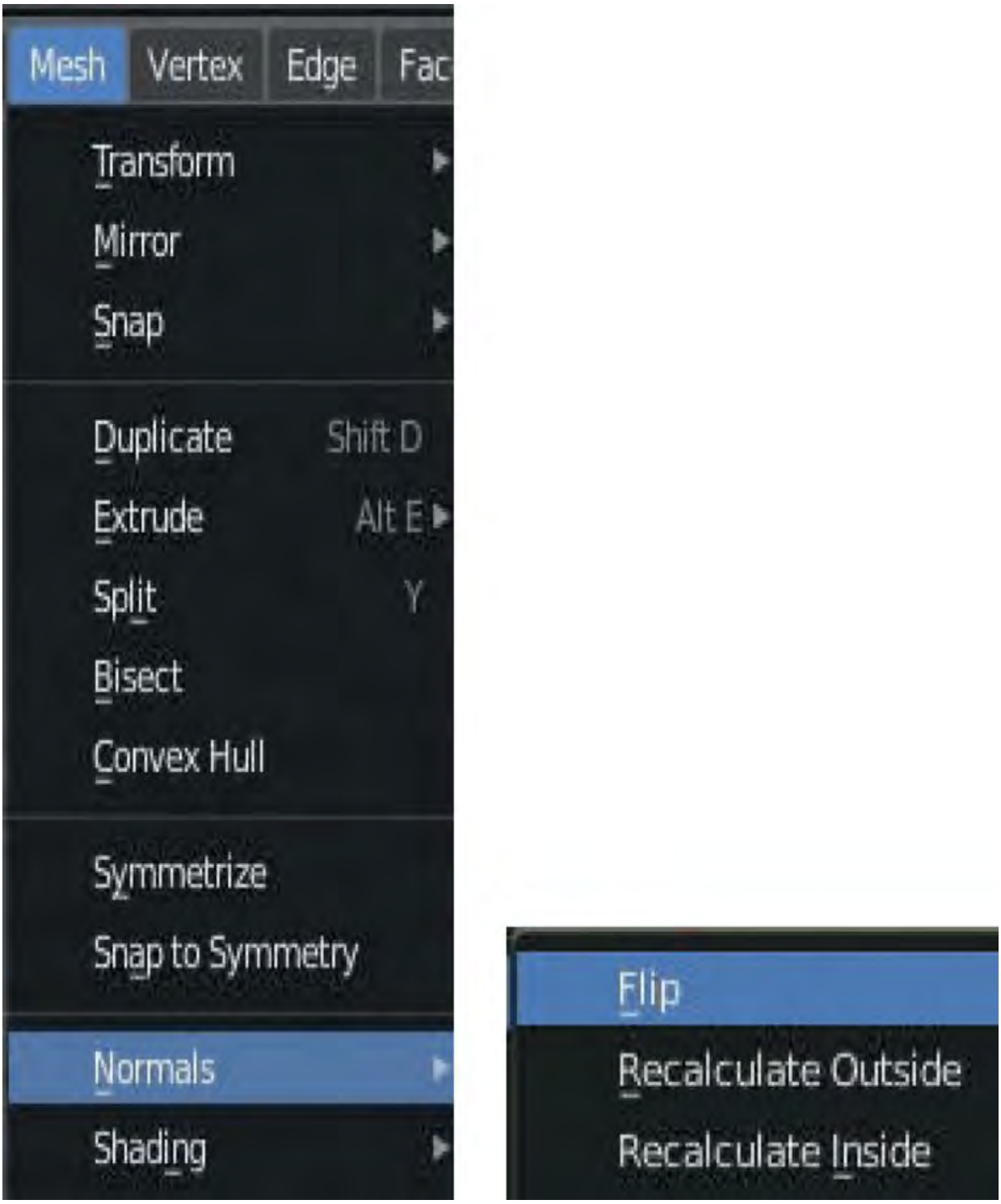

Mesh
Vertex
Edge
Fac
Transform
Mirror
Snap
Duplicate Shift D
Extrude Alt E
Split Y
Bisect
Convex Hull
Symmetrize
Snap to Symmetry
Normals
Shading
Flip
Recalculate Outside
Recalculate Inside

Figure 24.39

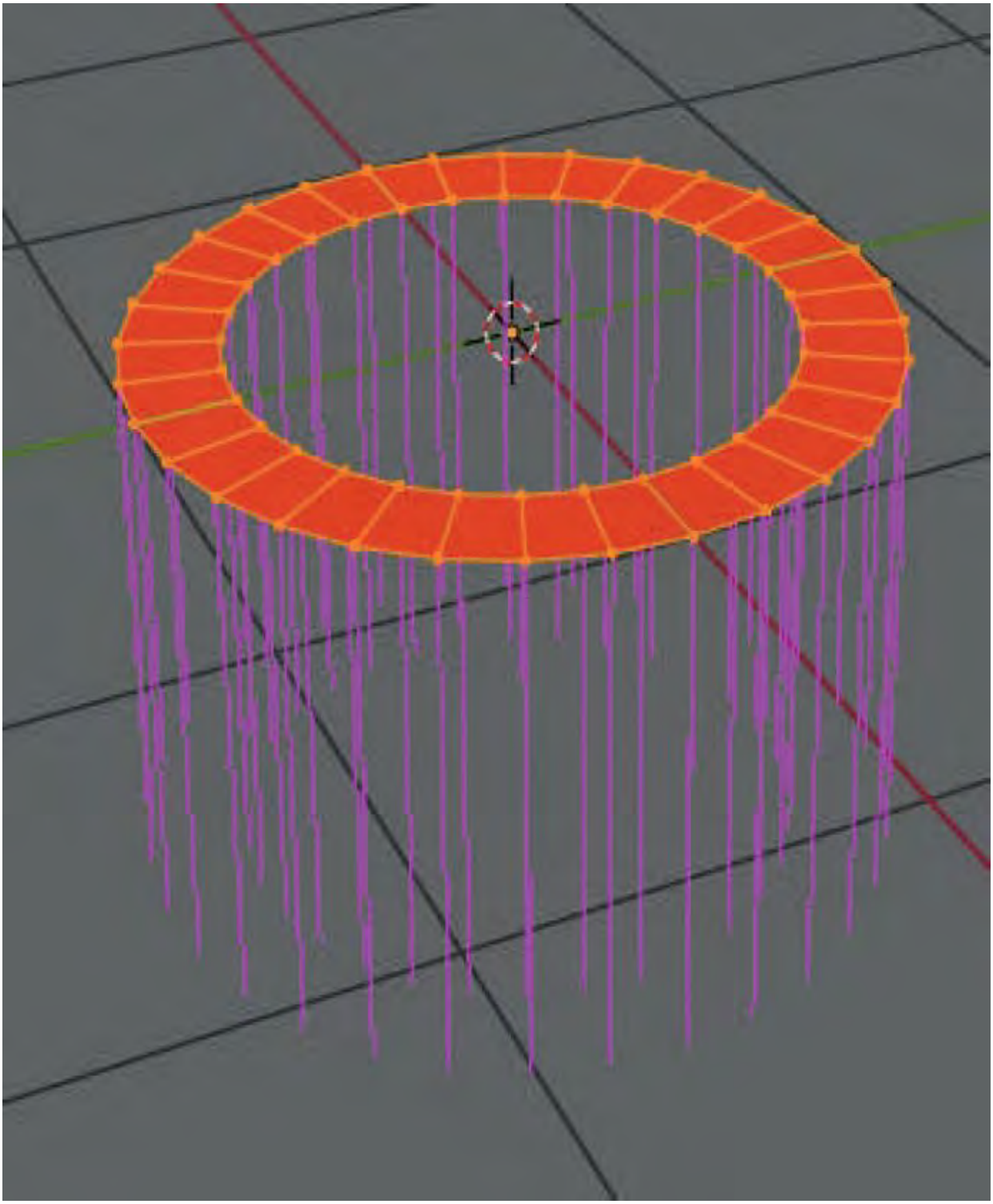


Figure 24.40

You may quickly change direction in the Edit Mode Screen Header, Mesh Button by selecting **Normals, Flip** (Figure 24.39).

Note: You can Not play the Animation in Edit Mode.

An example of creating a Particle Effect is as follows:

With the disk as shown in Figure24.40, flip the Normals into the upward direction. In Object Mode change the Emit From: Faces to **Vertices** in the **Source Tab** and **uncheck Random Order**. Change the **Lifetime** value in the Emission Tab to 200.

When the animation is replayed the Particles are Emitted progressively around the disk from the mesh Vertices and rise up forming a spiral configuration (Figure 24.41). The Particles are grouped in short columns.

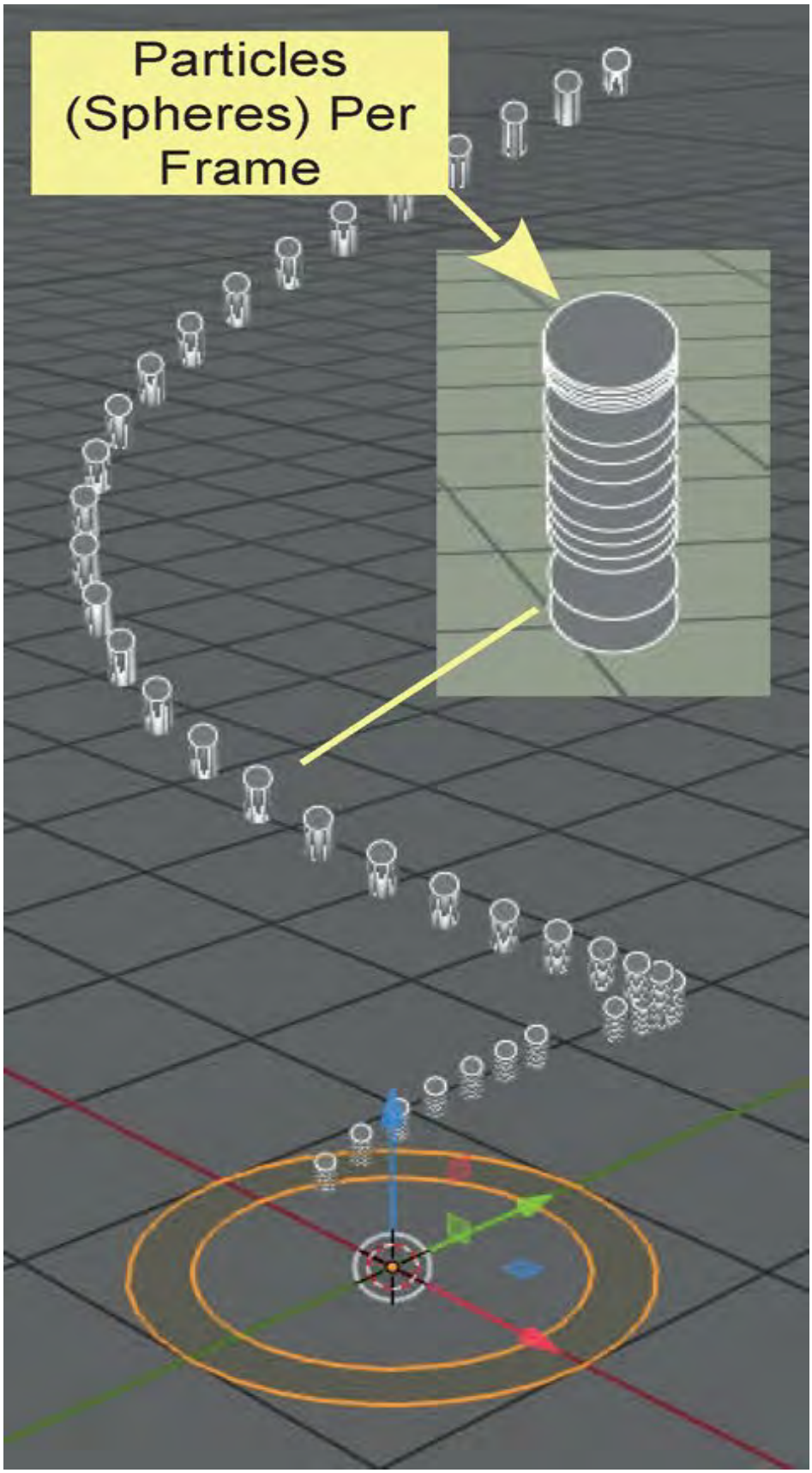

Particles
(Spheres) Per
Frame

Figure 24.41

In the **Velocity Tab** make the Emitter Geometry, **Normal: 0.000** and the **Object Alignment Y: 1.000**.

When the animation is replayed the Particles spiral on the Y Axis in the Scene (Figure 24.42).

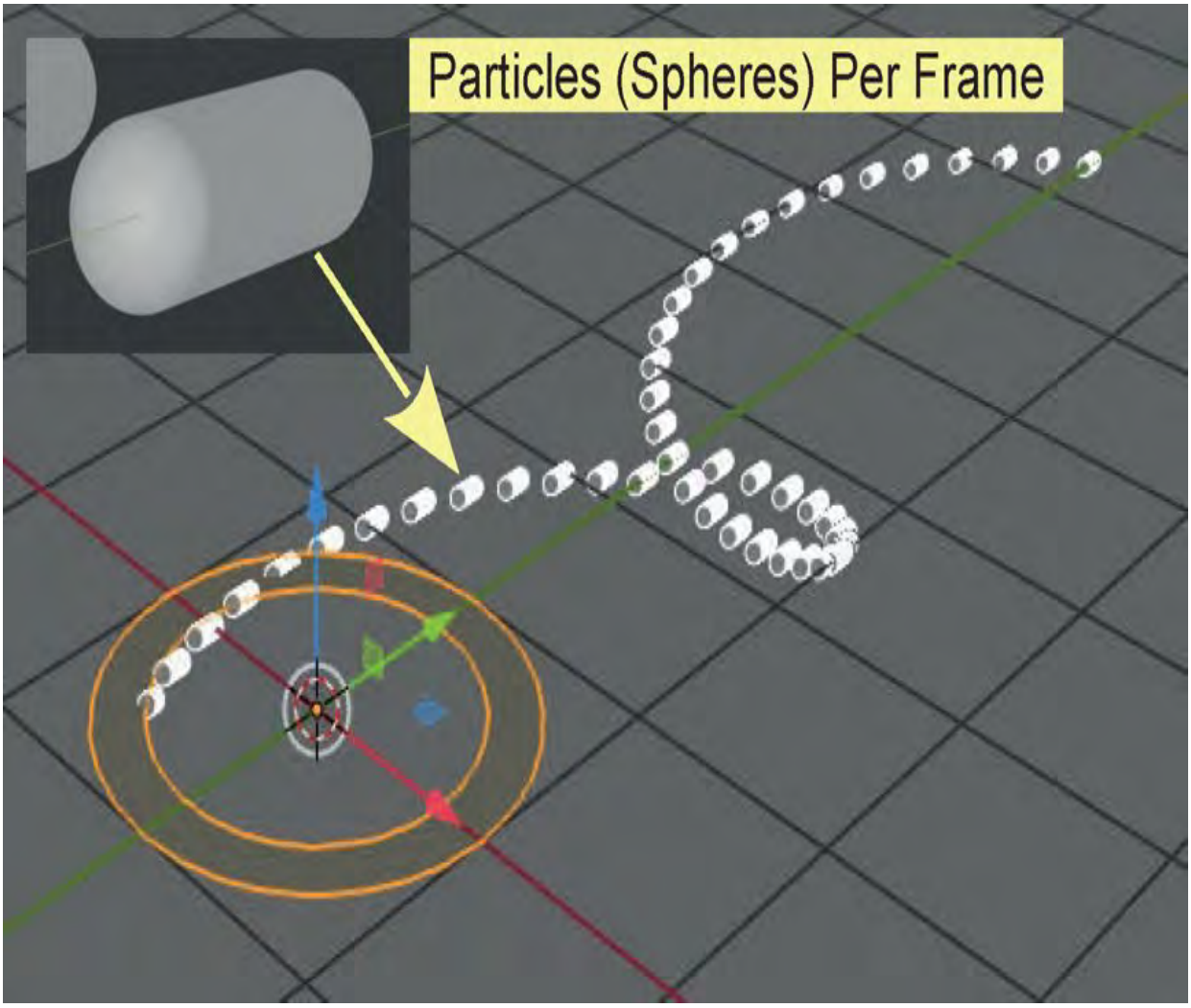


Figure 24.42

Up to this point Particles have been displayed in the 3D Viewport Editor as little white spheres in Object mode.

24.10 Particle Modifiers

In the **Properties Editor, Modifier Properties,** selection menu you will find the **Particle System Modifier** and the **Particle Instance Modifier**. Having gained a little knowledge in respect to the Particle System it is appropriate to mention these two Modifiers.

Particle System Modifier

Adding a **Particle System Modifier** to a selected Object merely adds a default Particle System. This is the same as going to the Particle Properties and clicking the **Plus Sign**. With the Modifier added you manipulate settings to achieve the desired result.

Particle Instance Modifier

The **Particle Instance Modifier** allows you to create an array of Objects mimicking the array of Particles which are being Emitted.

To demonstrate, set up a Scene in **Top Orthographic View** containing a Circle Object and a UV Sphere Object (scaled down) positioned as shown in Figure 24.43

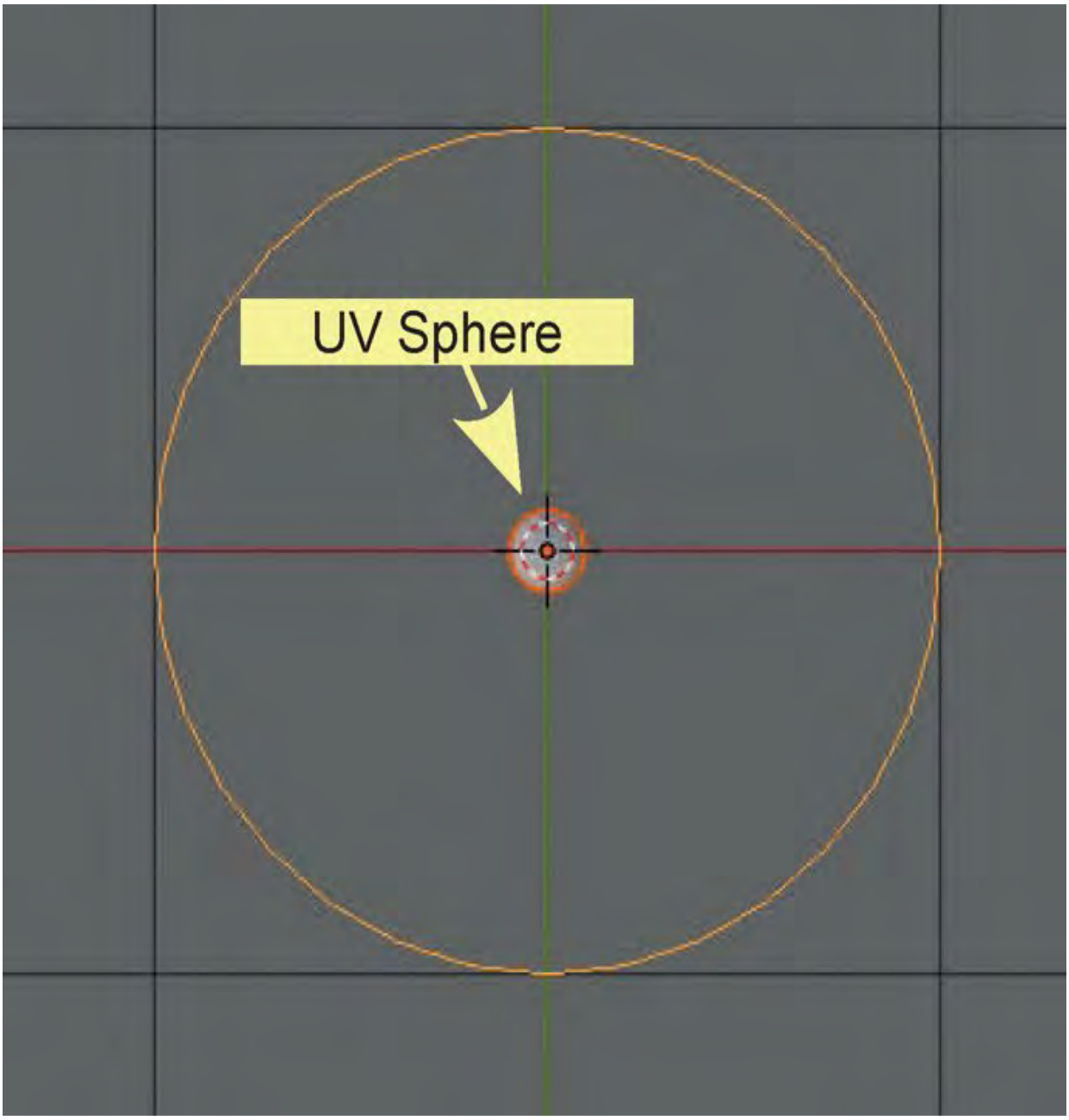


Figure 24.43

Turn Gravity off

Have the Circle selected and add a Particle System with values as follows:

In the Emissions Tab:	Number: 10
	Lifetime: 200
In the Source Tab	Emit From: Vertices
	Uncheck Random Order
In the Velocity tab:	Normal: 0.25 m/s

Do not play the animation at this point.

Deselect the Circle and select the UV sphere.

In the Properties Editor, Modifier Properties, Add a **Particle Instance Modifier** and enter **Circle** as the Object (the Circle being the Object with the Particle System applied).

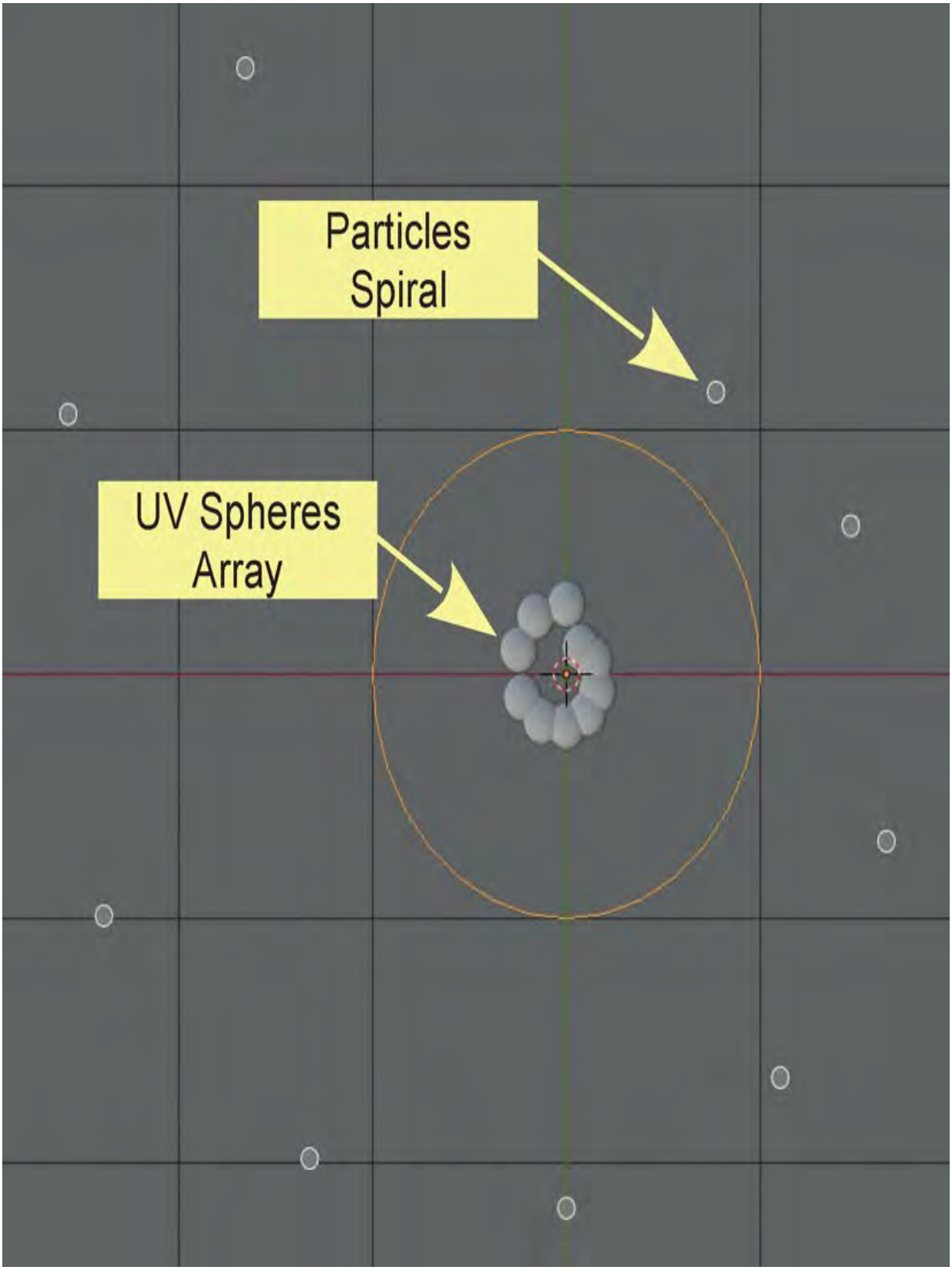


Figure 24.44

Play the animation in the Timeline window to see an array of Spheres generated in a spiral configuration mimicking the spiral of the Particles (Figure 24.44).

24.11 Particles Array

With the control of Particle Emission and the display of Particles as other Objects you can create arrays for effect when combined with the application of Materials and the addition of lighting . The following is an example.

In the default Blender Scene delete the Cube and add a UV Sphere. The default Scene has a single Point Light. Add a second Point Light and a Sun Light and position above and spaced around the UV Sphere

Add an **Ico Sphere,** set to smooth shading and scale 0.500. **Add a Material Color**. Park the Ico Sphere to one side of the Screen. You may hide the Ico Sphere from view by clicking the eye icon in the Collection in the Outliner Editor.

Select the UV Sphere (the Emitter Object) and add a Particle System. Increase the Emission Number to 30 000. In the Render Tab, Render as Object with the Ico Sphere as the Instance Object in the Object Tab. Deselect the UV Sphere.

Turn Gravity Off.

Add a Turbulence Force Field to the Scene (Shift + A Key – Force Field – Turbulence -Strength 5) (set the Strength value in the Properties Editor, Physics Properties, Settings Tab).

Play the animation to see the flow of Objects. Figure 24.45 shows the flow in **Material Preview Viewport Shading Mode** with the 3d Viewport Editor background (Gradient High/Off) color modified).

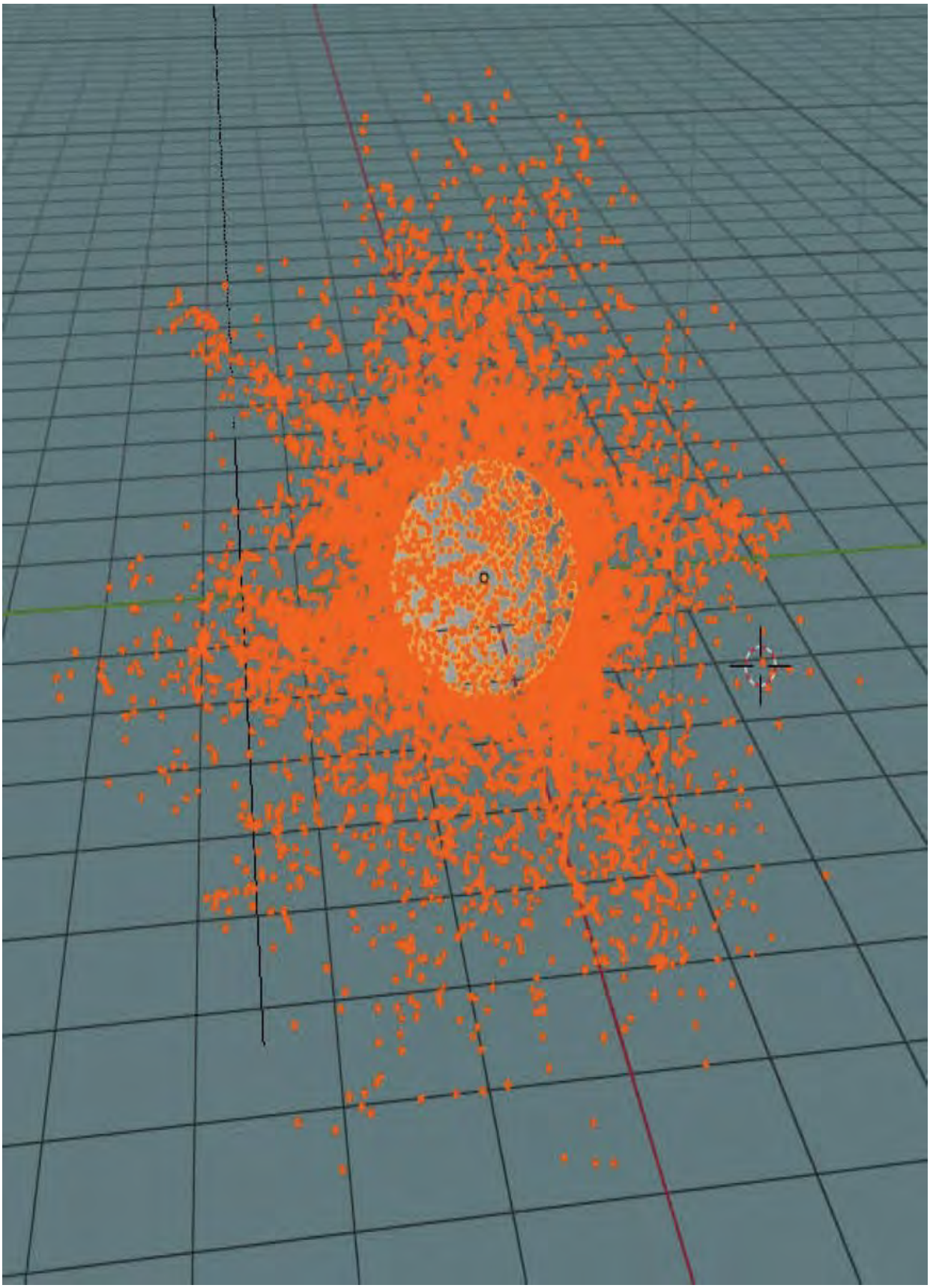


Figure 24.45

In the Particle Properties change Lifetime to 999 (forever) and the End value to 100. In the render Tab leave Scale: 0.050 but change Scale Randomness to 0.800.

Move the Timeline editor Cursor to frame 120 and rotate the Viewport to see the scatter of Ico Spheres.

Divide the 3D View Editor in two and have one in Material Preview Viewport Shading Mode and the other as Rendered Viewport Shading (Figure 24.46).

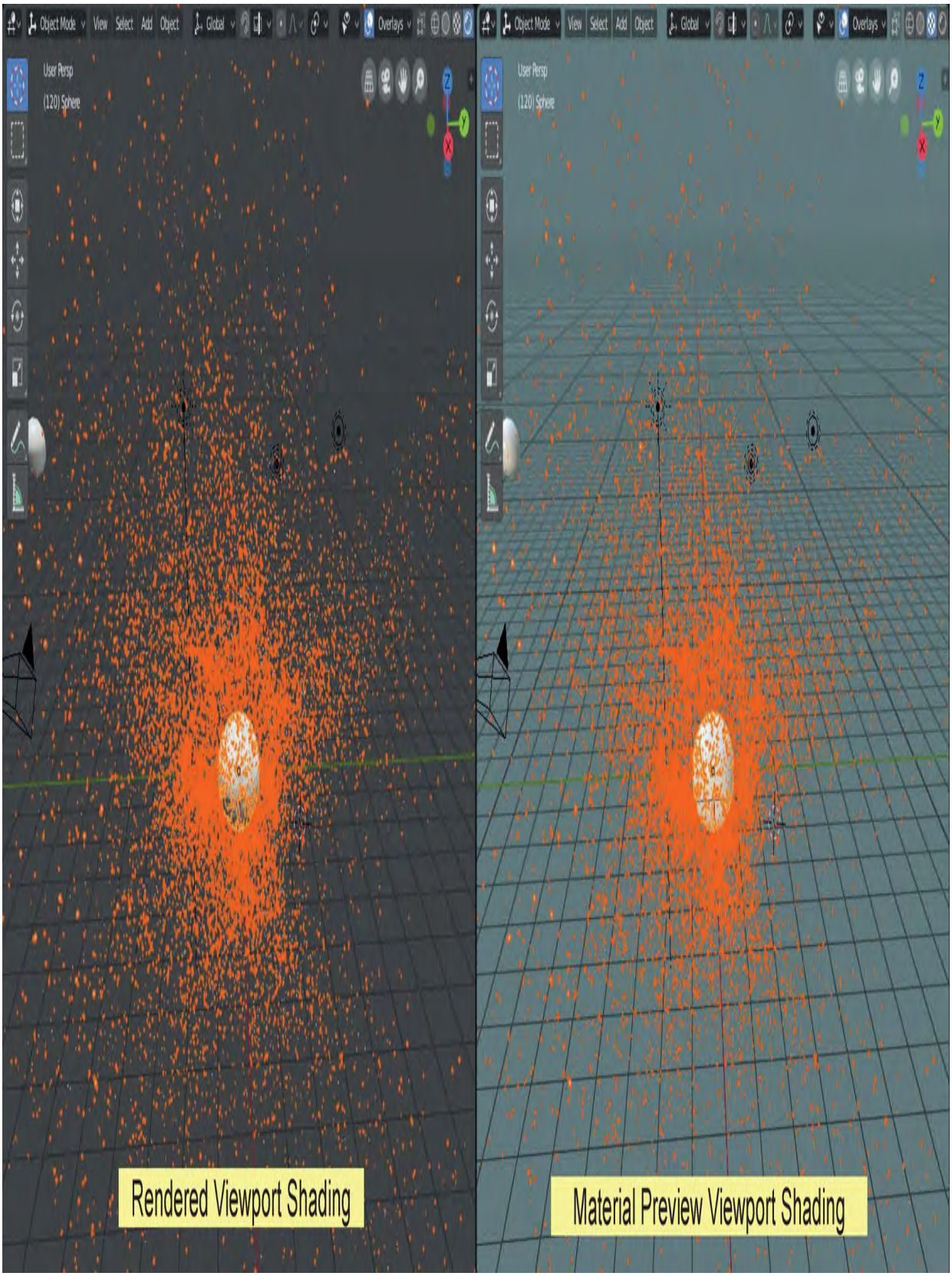

Object Mode
View
Select
Add
Object
Global
Overlays
User Persp
(120) Sphere
Rendered Viewport Shading
Material Preview Viewport Shading

Figure 24.46

Select the UV Sphere and add a Material and set to Smooth Shading.

Select the Ico Sphere and add a Material using the Node Arrangement shown in Figure 24.47.

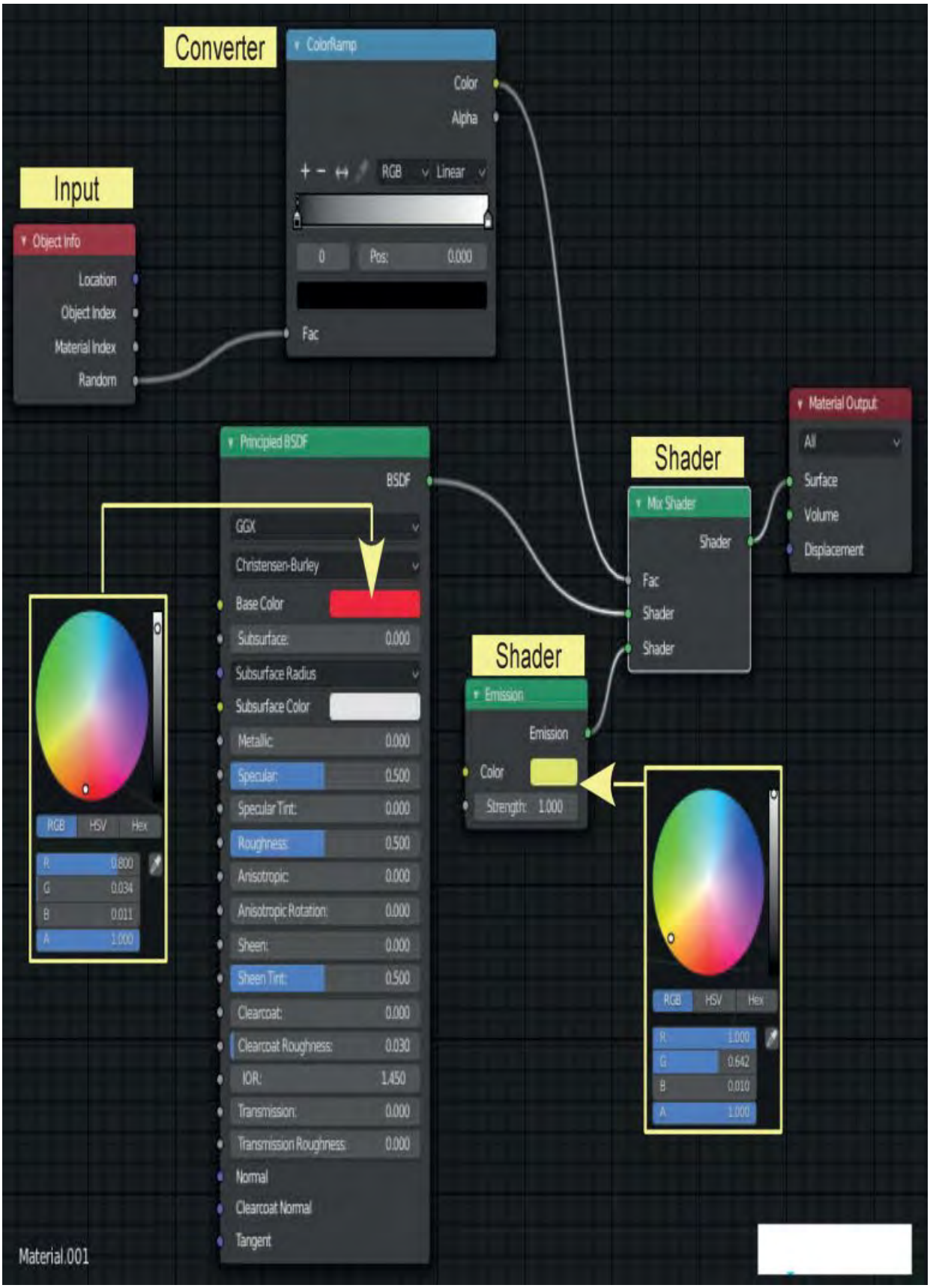

Converter
ColorRamp
Color
Alpha
RGB
Linear
0
Pos:
0.000
Fac
Input
Object Info
Location
Object Index
Material Index
Random
Material Output
All
Surface
Volume
Displacement
Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface:
0.000
Subsurface Radius
Subsurface Color
Metallic:
0.000
Specular:
0.500
Specular Tint:
0.000
Roughness:
0.500
Anisotropic:
0.000
Anisotropic Rotation:
0.000
Sheen:
0.000
Sheen Tint:
0.500
Clearcoat:
0.000
Clearcoat Roughness:
0.030
IOR:
1.450
Transmission:
0.000
Transmission Roughness:
0.000
Normal
Clearcoat Normal
Tangent
Shader
Mix Shader
Shader
Fac
Shader
Shader
Shader
Emission
Emission
Color
Strength:
1.000
RGB
HSV
Hex
R
0.800
G
0.034
B
0.011
A
1.000
RGB
HSV
Hex
R
1.000
G
0.642
B
0.010
A
1.000
Material.001

Figure 24.47

See the results in different Viewport Displays (Figure 24.48 over).

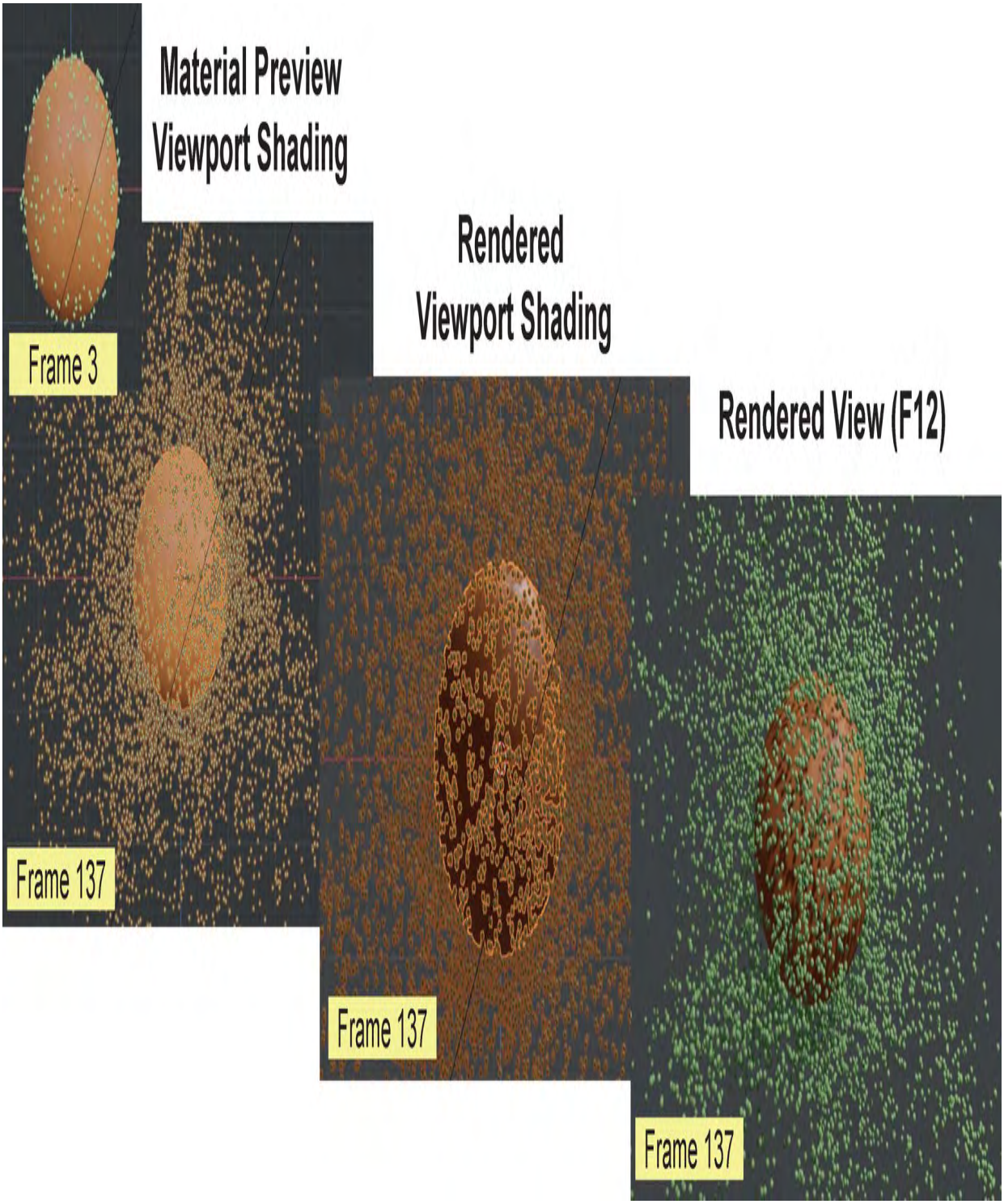


Figure 24.48

Viewport Displays

How results are viewed depends on the Viewport Shading employed.

24.12 The Viewnort Display Tab

How Particles display is controlled in the **Properties Editor, Particle Properties Viewport Display Tab**.

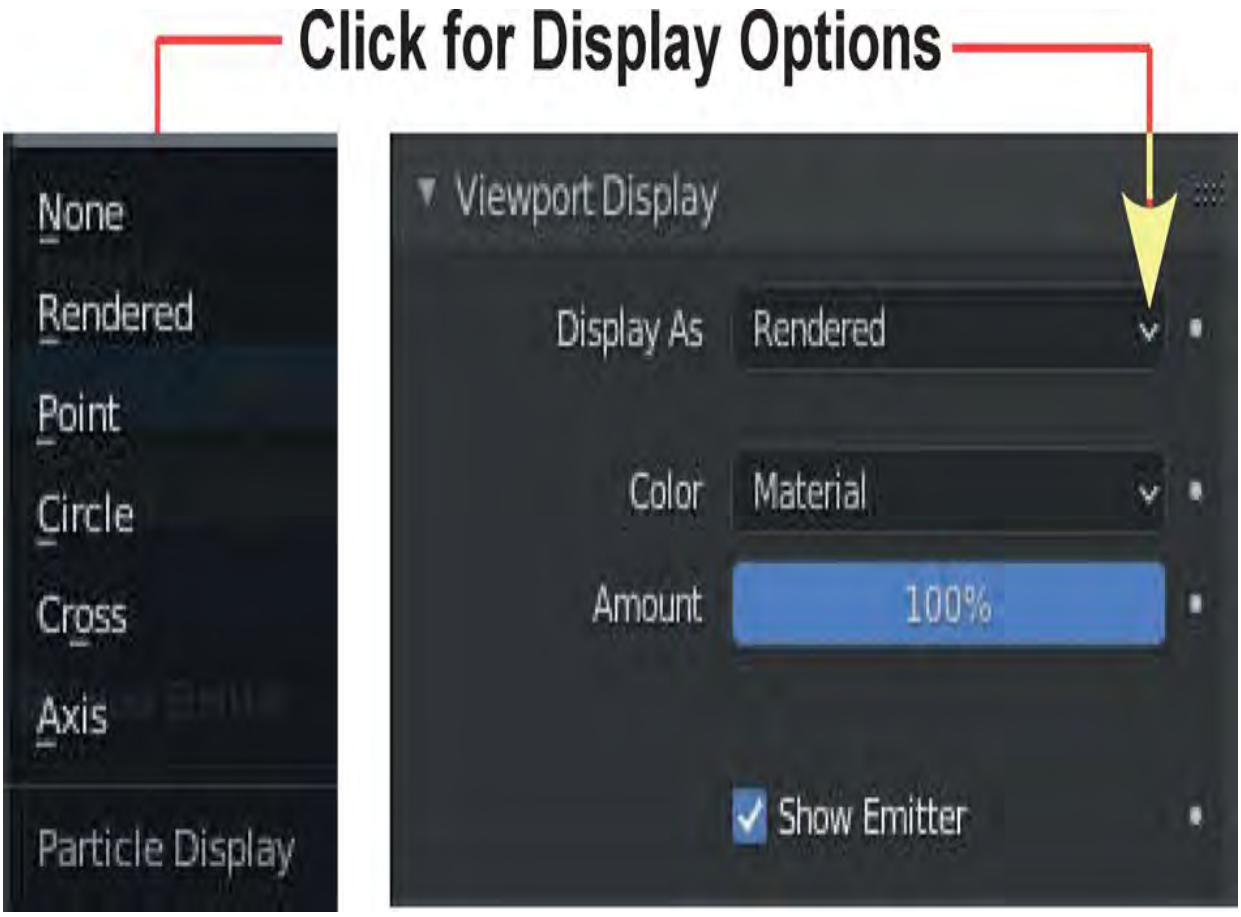


Figure 24.49

When multiple Particle Systems are in play it is advantageous to display Particles from one system differently to another. This is not to be confused with what takes place in a Rendered Image of a Scene. Particles themselves

do not Render. In Figure24.48 above, the Particles are Rendered as the Ico Sphere (the Instance Object). In the default 3D Viewport Editor the Particles display as white spheres with the Display set as Rendered. They also display as spheres with **Display As: Point** is set. Alternative displays are shown in Figure 24.50.

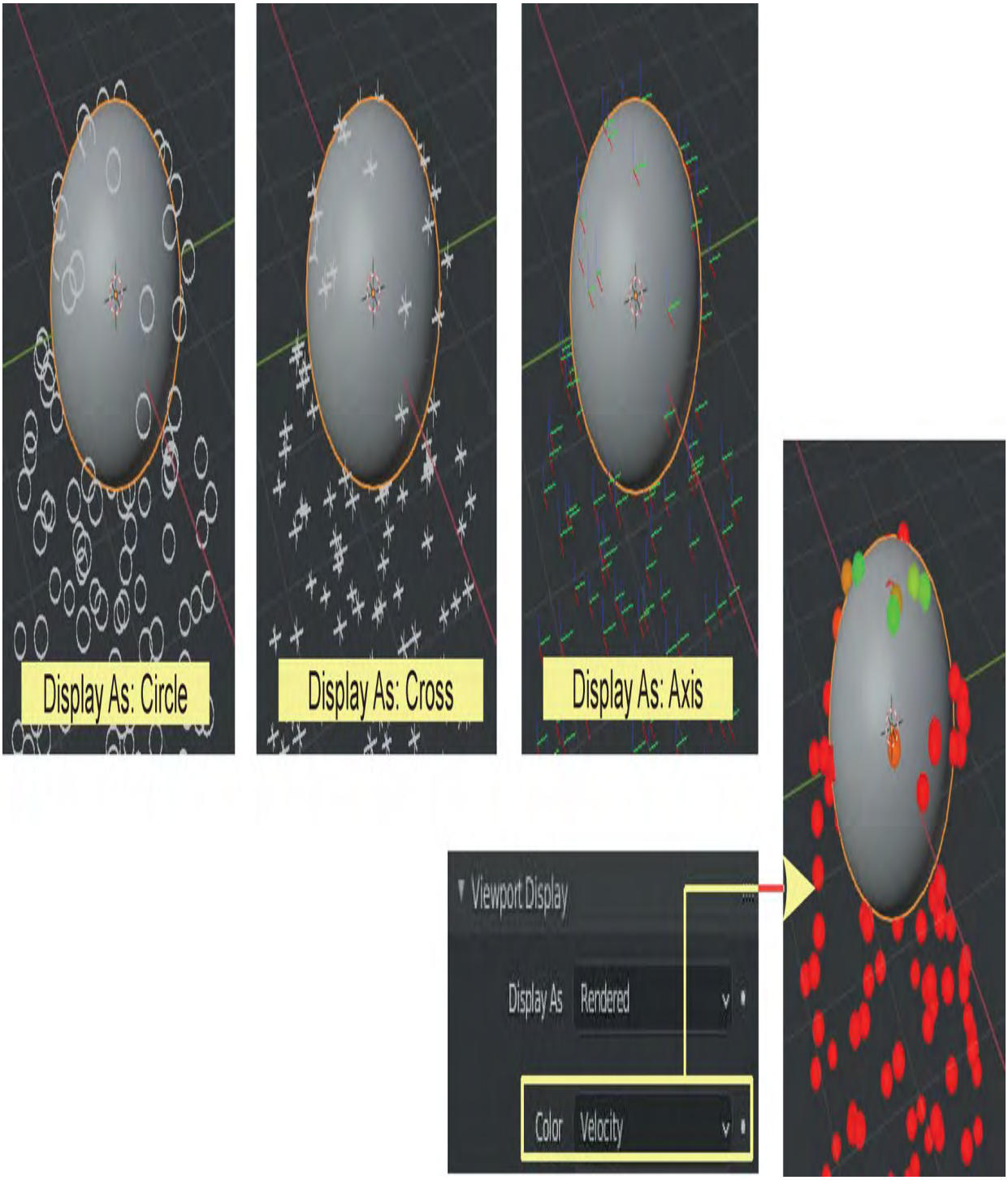


Figure 24.50

24.13 Particle Interaction

Particles can interact with other Objects and be affected by Forces like Wind, Turbulence and Drag. Particles can bounce off other Objects and act like sparks or droplets. To show how these features work, set up a Scene with a UV Sphere positioned above a Plane as shown in Figure 24.51 (the Plane is scaled up three times).

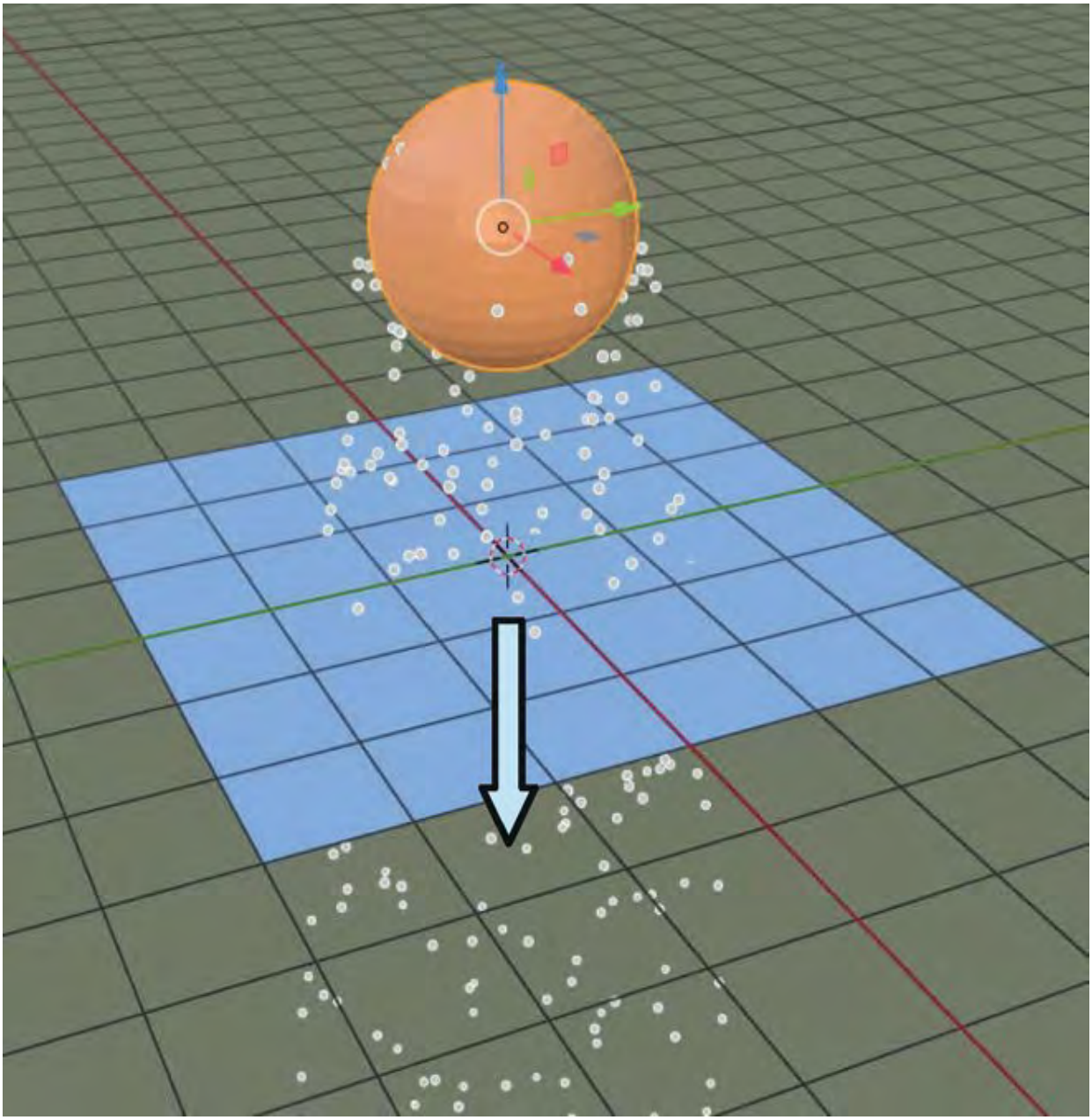


Figure 24.51

With the UV Sphere selected, go to the **Properties Editor, Particle Properties** and add a **Particle System** (to the UV Sphere).

In the **Emission Tab,** set the **End** value to 100 and in the **Velocity Tab,** set the **Object Aligned: Z value** to –3.000 (Emitter Object gives the Particles a starting velocity -3 down).

In the **Timeline Editor** press the Play button. You will see the Particles fall and pass through the Plane (Figure 24.51).

To stop the Particles passing through the Plane, select the **Plane** and go to the **Properties Editor, Physics Properties** (Figure 24.52). Select **Collision** and replay the animation **(Remember** you must be at frame 1 before you replay).

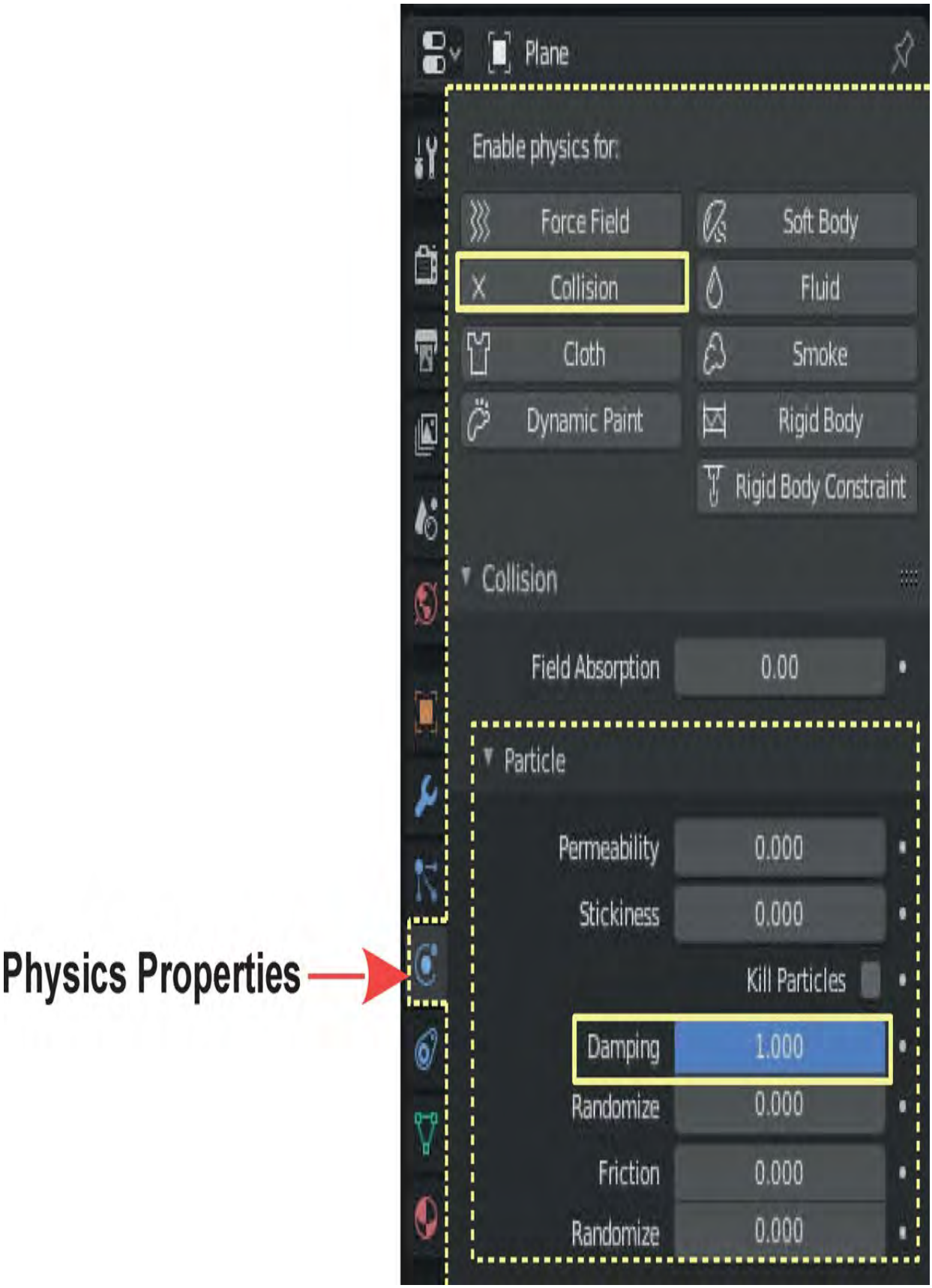

Physics Properties
Plane
Enable physics for:
Force Field
Soft Body
Collision
Fluid
Cloth
Smoke
Dynamic Paint
Rigid Body
Rigid Body Constraint
Collision
Field Absorption
0.00
Particle
Permeability
0.000
Stickiness
0.000
Kill Particles
Damping
1.000
Randomize
0.000
Friction
0.000
Randomize
0.000

Figure 24.52

The Particles bounce up from the surface of the Plane (Figure 24.53).

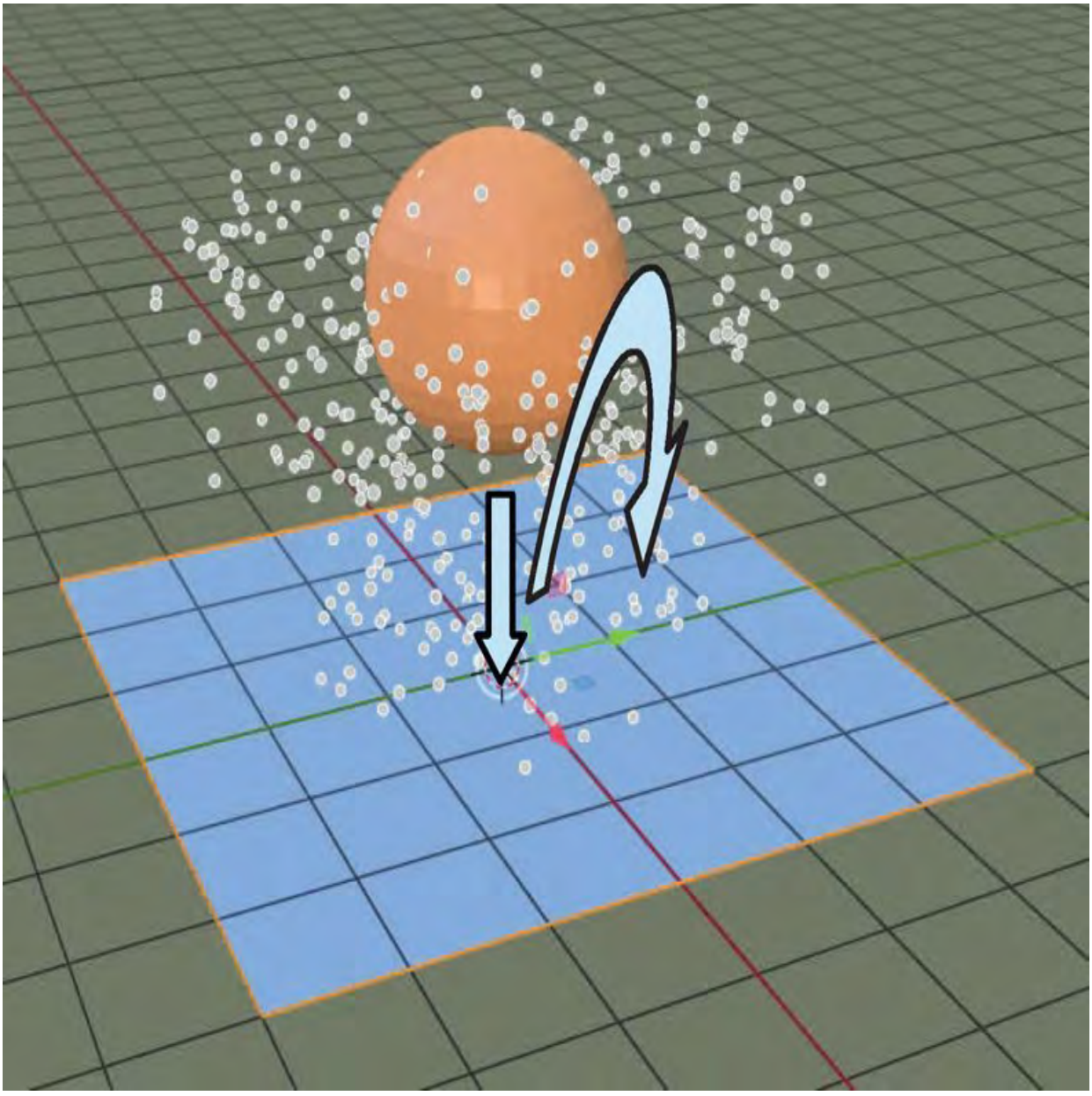


Figure 24.53

By increasing the **Particle Damping: Factor** value in the **Particles Tab** to 1.000 (Figure 24.52), the Particles will land on the Plane but they will no

longer bounce; they will just slide on the surface (Figure 24.54).

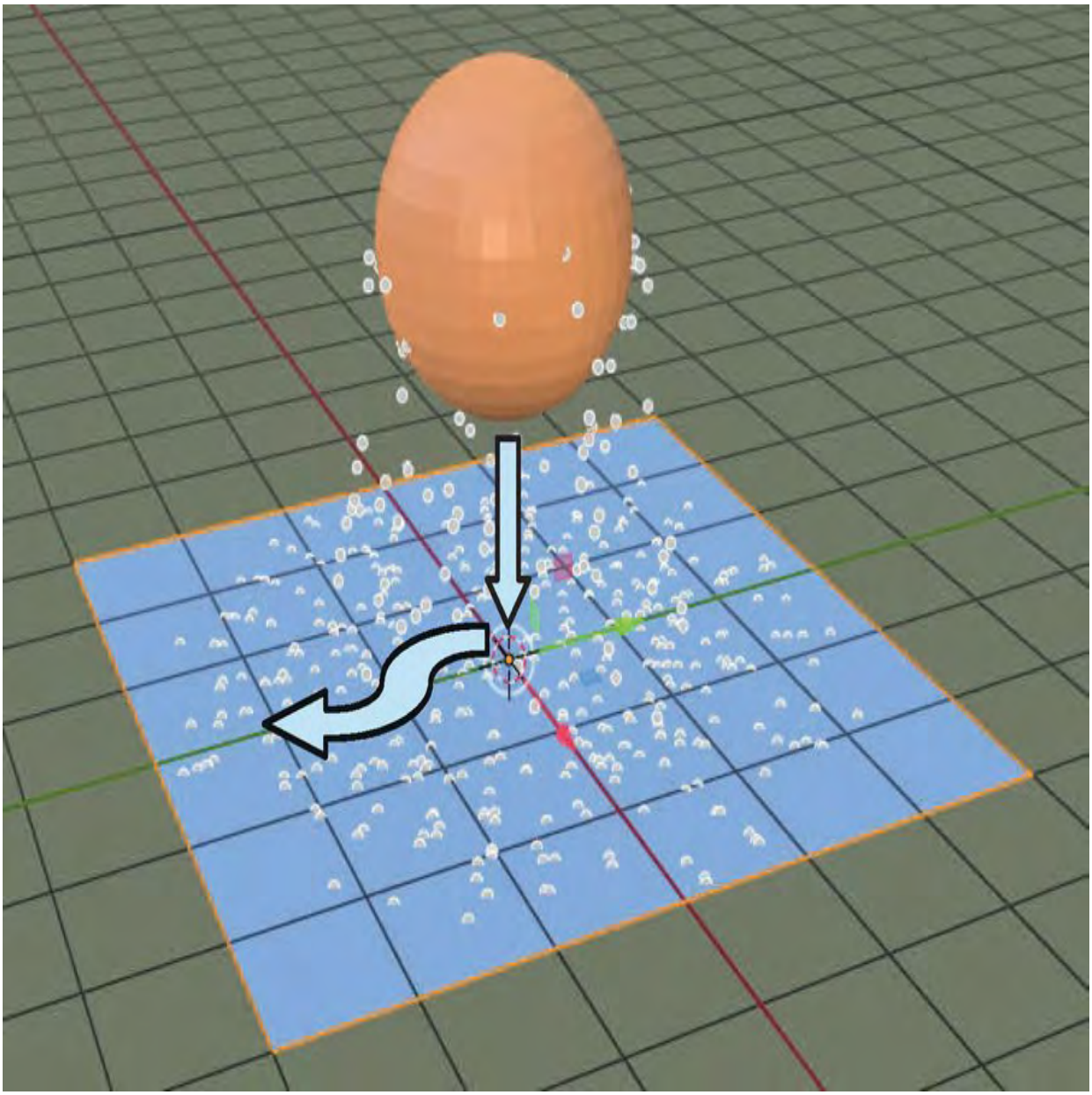


Figure 24.54

24.14 Wind Force Effect

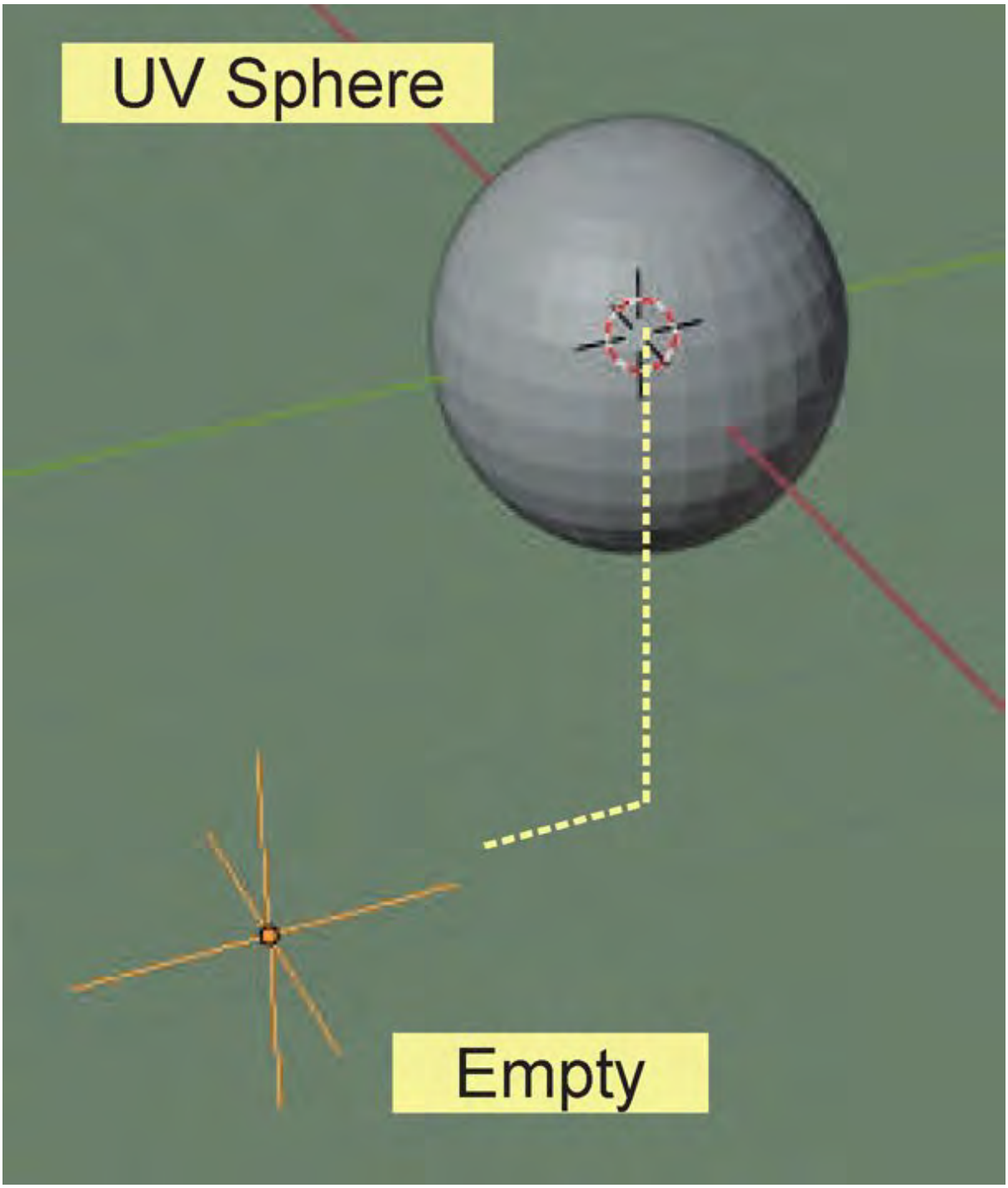
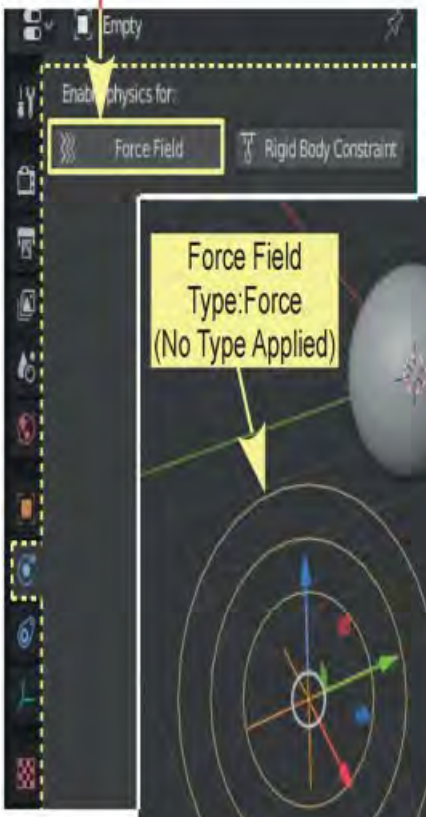


Figure 24.55

Particles can be influenced by a simulated **Wind Force**. To create a wind effect, you have to place an Object in the scene and assign a wind Force to it. The **Empty** Object is ideal for this since it doesn't render.

Set up a Scene with an Empty Object located below a UV Sphere and off to the side (Figure 24.55). Add the default Particle System to the UV Sphere. The UV Sphere will emit Particles that will fall downwards since the Gravity box is ticked in the Scene tab.

With the **Empty** selected in the 3D Viewport Editor, go to the **Properties Editor, Physics Properties** and click on the **Force Fields button** (Figure 24.56).



Clicking the **Force Field** button applies a Force Field to the Empty Object and displays the controls for the Force Field in the Properties Editor.

Force Type: Wind applied to the Empty

In the **Force Field Tab** click on **Type** and select **Wind** in the menu that displays.

Force Type: Wind applied

Figure 24.56

With the default **Force Type: None** applied there is no effect on the Particles which emit from the UV Sphere when an animation is played.

Force Type: Wind applied

Note: With the **Force Type: Wind** the 3D Viewport Editor shows a series of concentric rings. The orange arrow indicates the direction of the Force and the Yellow arrow is a control handle for adjusting the strength of the Force. Strength may also be adjusted in the Properties Editor.

The objective in this demonstration is to have the Wind Force blow Particles descending from the UV Sphere to the right along the Y Axis of the Scene.

With the Empty selected press the R Key (rotate) + X Key (rotate about the X Axis). Increase the Strength value to approximately 7.6.

With the Empty positioned as shown in Figure 24.57, Particles being emitted (when the Animation is played) descend due to the force of Gravity and are deflected in the positive Y Axis direction.

3D Viewport Editor
Properties Editor
Right Orthographic View
Y Axis
Strength
Falloff: Min – Max Distance
Empty
Force Field
Rigid Body Constraint
Force Fields
Type
Wind
Settings
Shape
Line
Strength
7.600
Flow
0.000
Affect
Location
Rotation
Noise Amount
0.000
Seed
6
Absorption
Wind Factor
0.000
Falloff
Shape
Sphere
Z Direction
Both Z
Power
0.000
Min Distance
1.4 m
Max Distance
3.46 m

Figure 24.57

Note: The Wind Force **Strength** is indicated by the seperation of the orange rings. The Wind Force effect on the descending Particles is influenced by the **Min/Max Distance** values set in the Falloff Tab.

24.15 Boids Particles

Boids Particle Systems are used to simulate flocks, swarms, herds and schools of various kinds of animals or anything that acts with similar behaviour. **Boids** Particle Systems are of **Type: Emitter** with **Boids Physics** applied. **Note:** The Physics is applied in the Properties Editor, Particle Properties NOT in the Physics Properties.

Boids Particles in one Particle System can react to Particles in another system or they can react to Particles within their own system. Boids are given **rules of behaviour,** which are listed in a Stack. The rules at the top of the Stack take precedence over rules lower down, but the Stack may be rearranged once it is written. **Properties Editor**

Since only a certain amount of information is evaluated, if the memory capacity is exceeded, rules lower down the Stack are ignored.

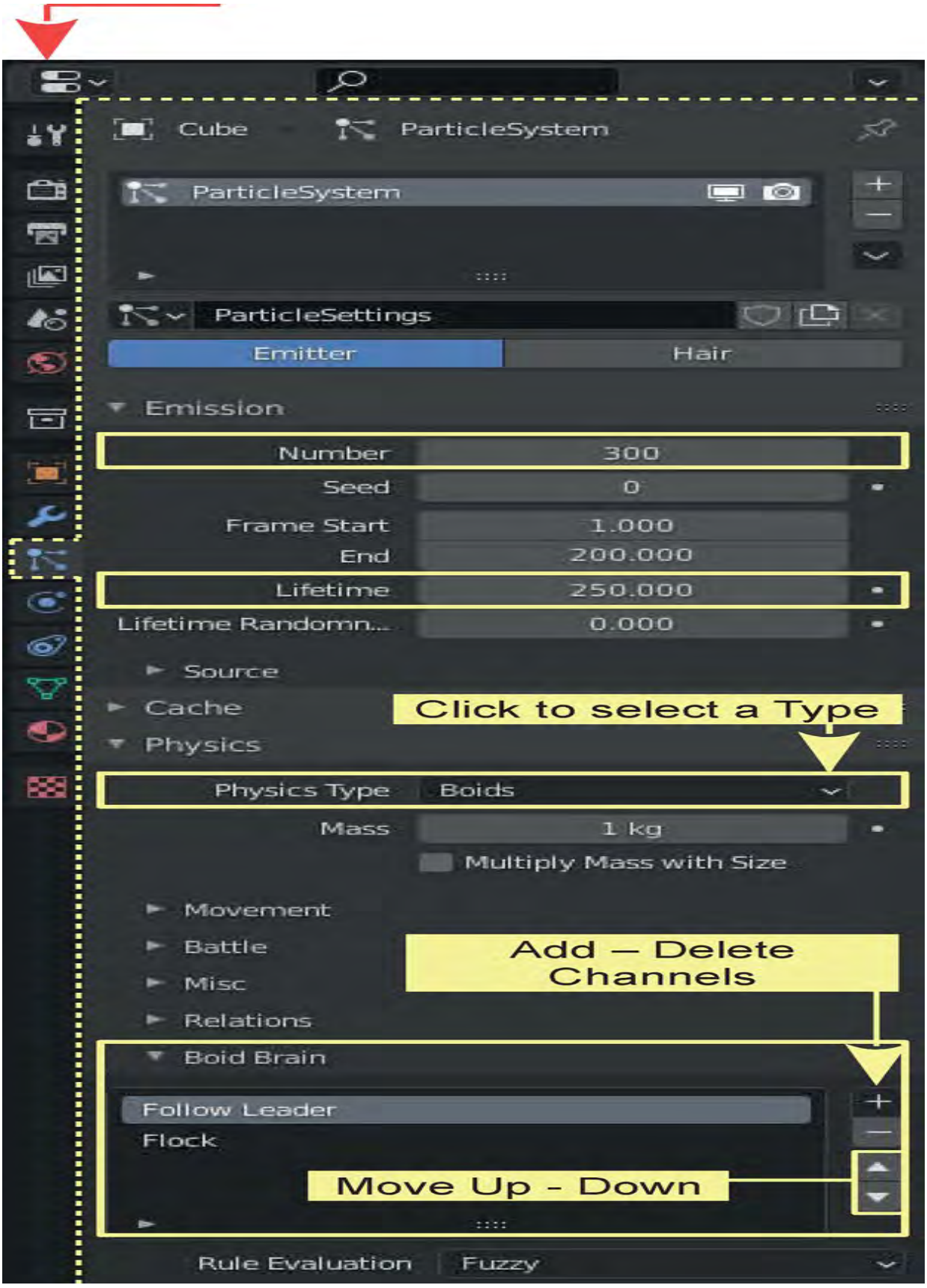

Cube
ParticleSystem
ParticleSystem
ParticleSettings
Emitter
Hair
Emission
Number
300
Seed
0
Frame Start
1.000
End
200.000
Lifetime
250.000
Lifetime Randomn...
0.000
Source
Cache
Click to select a Type
Physics
Physics Type
Boids
Mass
1 kg
Multiply Mass with Size
Movement
Battle
Misc
Relations
Add – Delete Channels
Boid Brain
Follow Leader
Flock
Move Up - Down
Rule Evaluation
Fuzzy

Figure 24.58

The procedure for setting up **Boids Particle Systems** will be demonstrated with the following examples.

Example 1: A Flock of Birds

In this example the Particles will act like a flock of birds but won't actually look like birds.

Open a new Blender Scene with the default Cube. The Cube will be the **Particle Emitter**. In the **Properties Editor, Particles Properties,** add a new **Particle System**. Leave all the settings with their default values, except for the following:

- **Emission Tab**
  - – Number: 300 (Have a small flock.)
  - – Lifetime: 250 (The default animation length in the timeline.)
- **Physics Tab**
  - – Select Physics **Type:Boids**.
- **Boid Brain Tab** (in the Physics Tab)
- With **Separate** highlighted, hit the **minus** sign to delete it. Click on the **plus** sign at the RH side of the window to display a selection menu for **Boids rules** and select **Follow Leader**. Click on the up arrow below the minus sign to move Follow Leader to the top of the Stack.

You have instructed the Particles to follow the leader while flocking together. You will now give the Particles a leader to follow.

Deselect the Cube in the 3D Viewport and add an **Empty**.

The Empty is a location point that can be animated to move in the Scene but does not render. Select the Empty and move it to the side. Animate the Empty to move across the Screen (see Chapter 20 for a refresher on Animation. Say

11 grid spaces in 140 Frames (gives PS time to catch up). Deselect the Empty and select the Cube.

In the **Boid Brain Tab,** make sure **Follow Leader** is highlighted (gray) (Figure 24.59).

Below the stack panel Click in the panel and select **Empty** from the menu that displays. You have instructed the Particles to follow the Empty. When the animation is played Particles Emitted from the Cube, head towards the Empty, and attempt to follow it as it moves across the Screen (Figure 24.60).

Figure 24.59

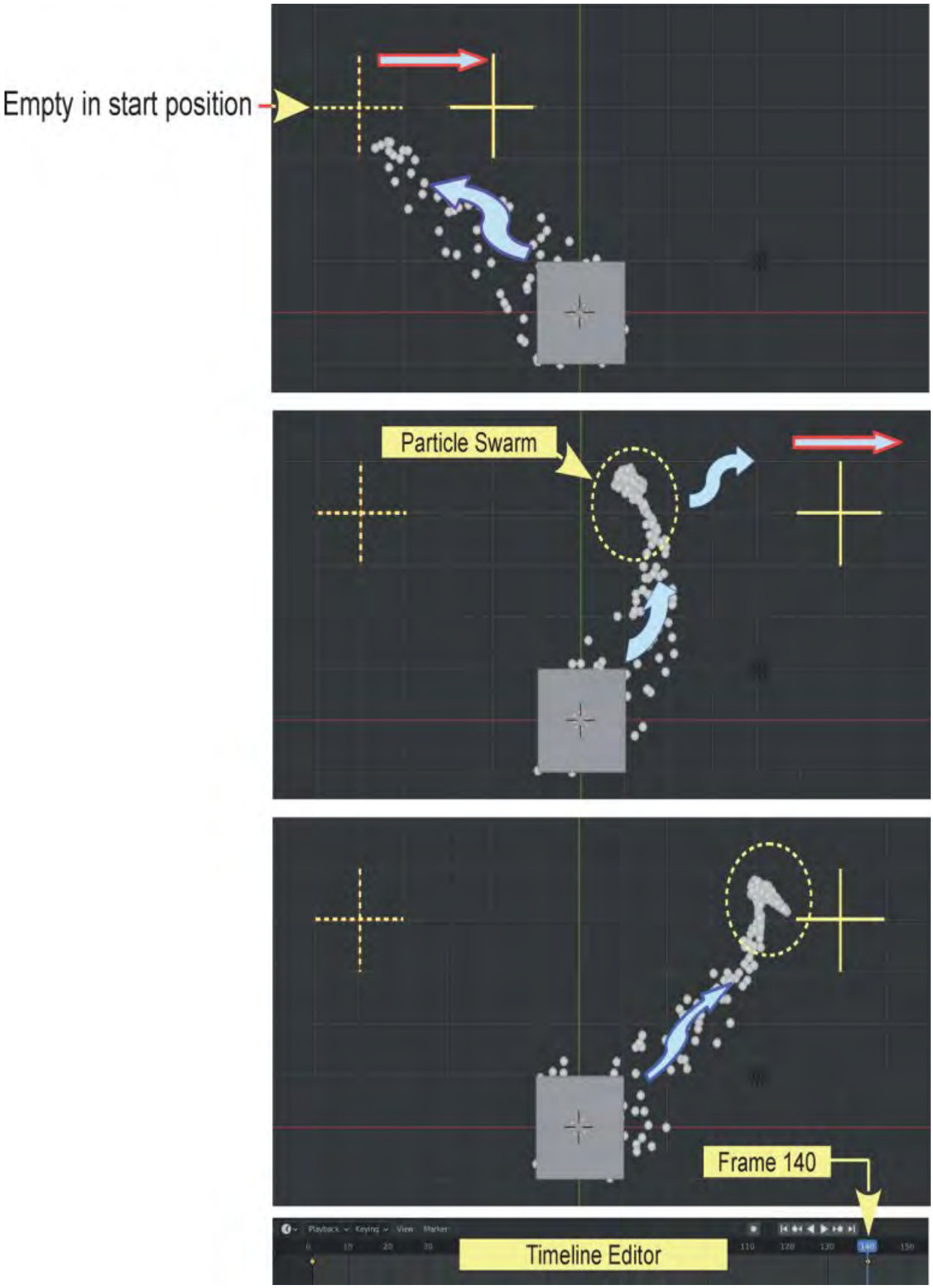

Empty in start position
Particle Swarm
Frame 140
Timeline Editor

Figure 24.60

Particles emitted from the Cube head towards where the Empty started from.

The Particles clump into a swarm and chase the Empty. Particles continue being emitted and follow the swarm The Empty stops moving at Frame 140 but Particles continue to be emitted and join the swarm.

Note: With a high Particle count your computer may crash if the computer lacks power.

Example 2: Directing Movement

This example will demonstrate how to direct Particles to move from one Object to another.

Start a new Scene and add a second Cube Object. Note that the default Cube is named **Cube** (see the upper left side of the 3D Viewport Editor) and the new Cube is named **Cube.001**.

Position the Cubes as shown in Figure 24.61, scaling the new Cube down.

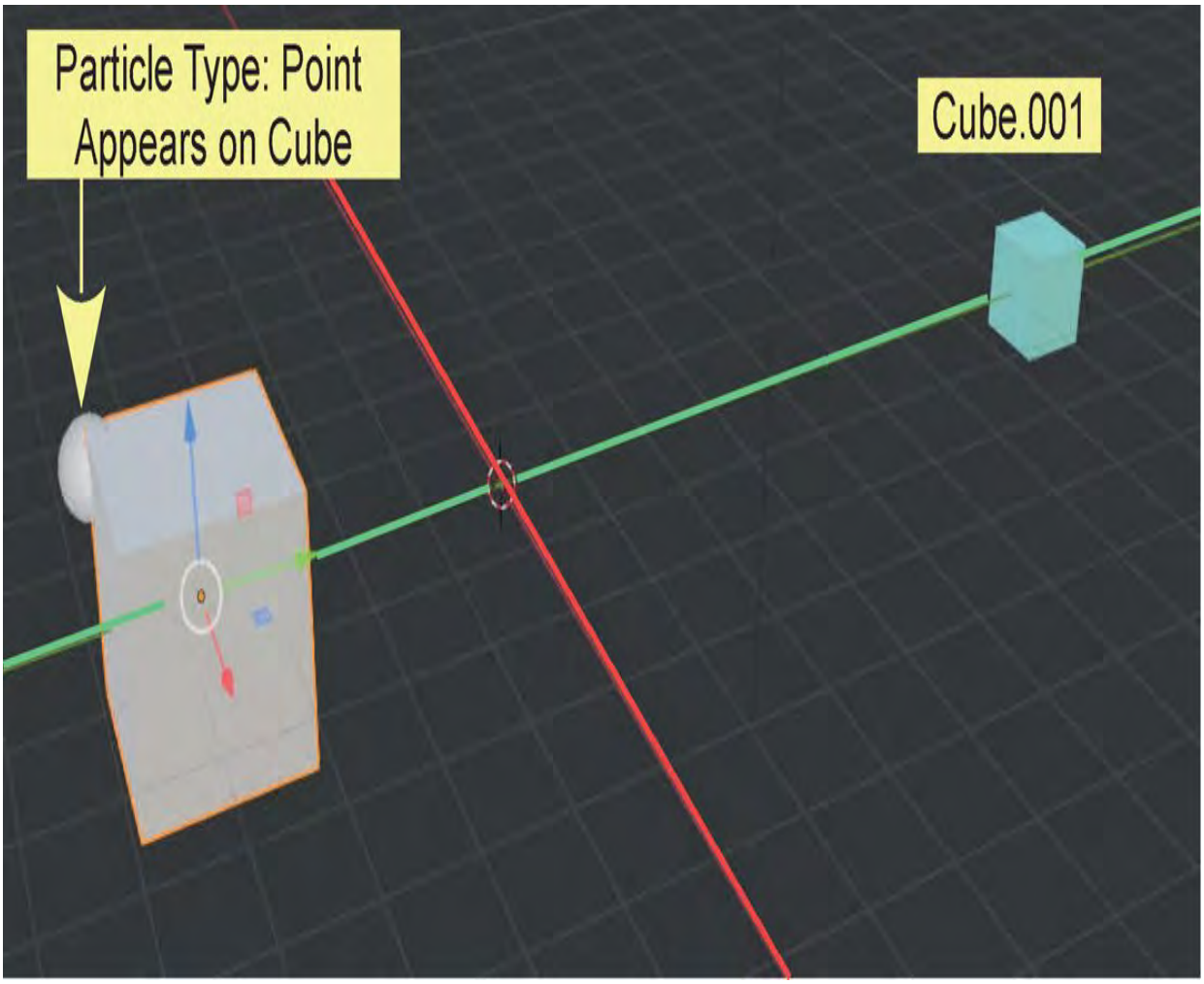


Figure 24.61

Select **Cube** and add a **Particle System** with **Physics, Boids Physics**. In the **Emission Tab,** reduce the **Number** value to **10** and set the **Lifetime** value to **1500;** you want to keep the number of Particles low and have them visible for a fair amount of time in the animation.

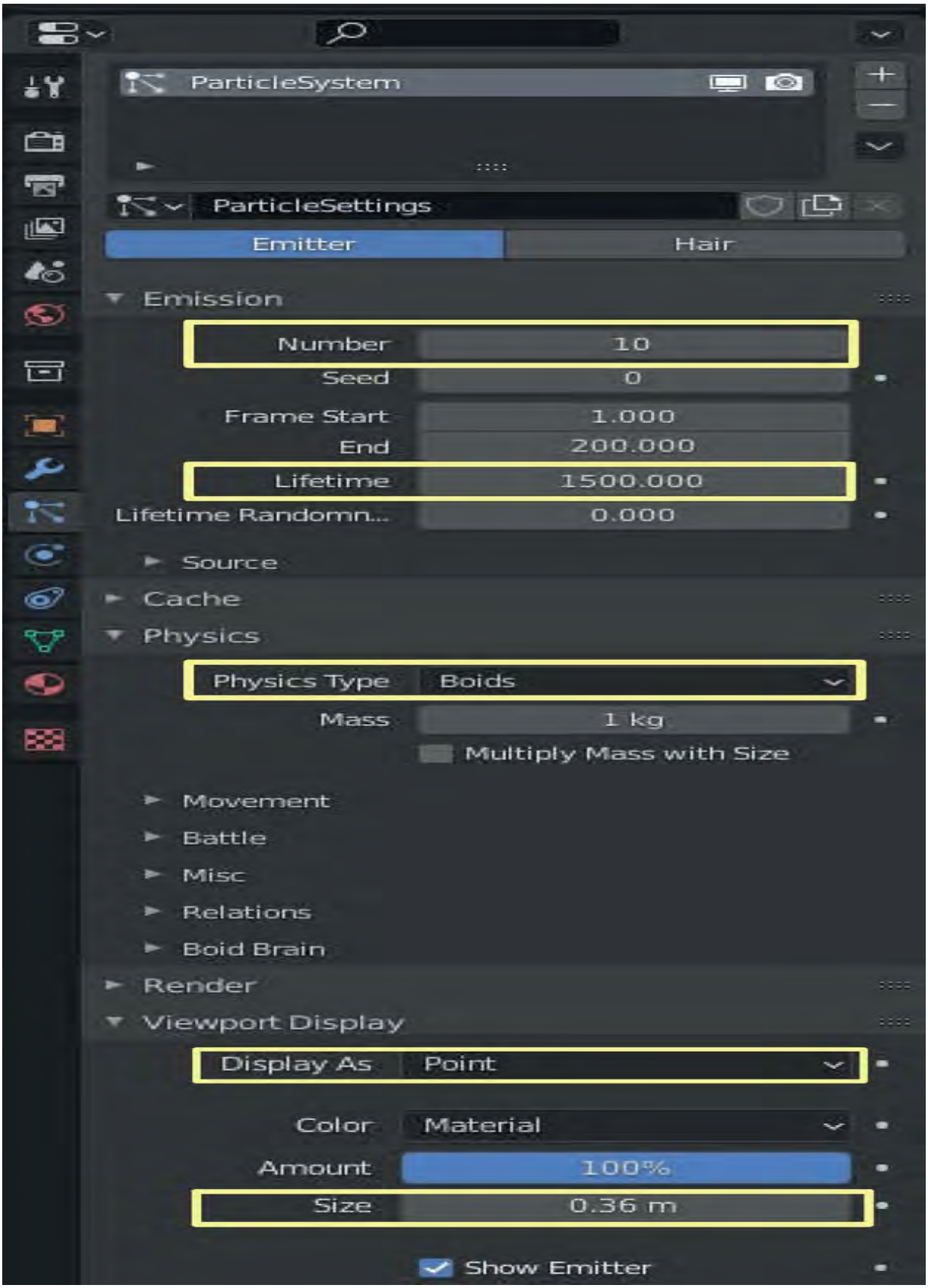

ParticleSystem
ParticleSettings
Emitter
Hair
Emission
Number
10
Seed
0
Frame Start
1.000
End
200.000
Lifetime
1500.000
Lifetime Randomn...
0.000
Source
Cache
Physics
Physics Type
Boids
Mass
1 kg
Multiply Mass with Size
Movement
Battle
Misc
Relations
Boid Brain
Render
Viewport Display
Display As
Point
Color
Material
Amount
100%
Size
0.36 m
Show Emitter

Figure 24.62

Go to the **Timeline Editor** and set the animation **End** value to **1500 Frames**.

In this example the **Particle, Viewport Display** will be **Point**.

In the **Particles Properties, Viewport Display Tab,** set **Display As: Cross** with the default **Size** value **0.36m;** you will see a sphere appear on the Cube (Figure 24.61).

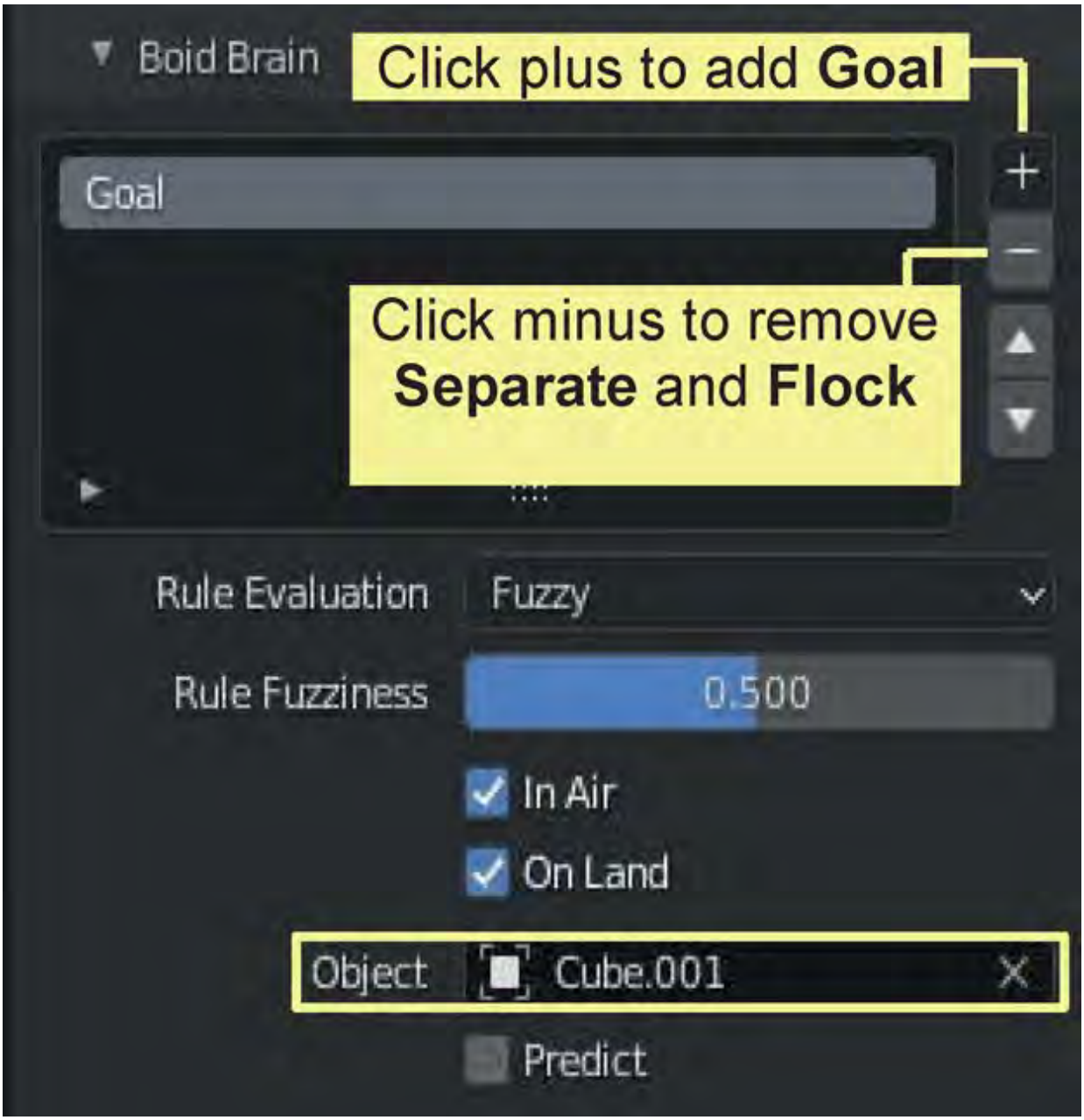


Figure 24.63

In the **Boid Brain Tab,** remove **Separate** and **Flock** and add **Goal** (Figure 24.63).

Click in the **Object Panel** and select **Cube.001 -** this tells the Particles Emitted from the original cube to go and find the target which is **Cube.001**.

Play the animation to see the result. Spheres emitted from **Cube** migrate across the Screen and accumulate on the target Cube.001(Figure 24.64).

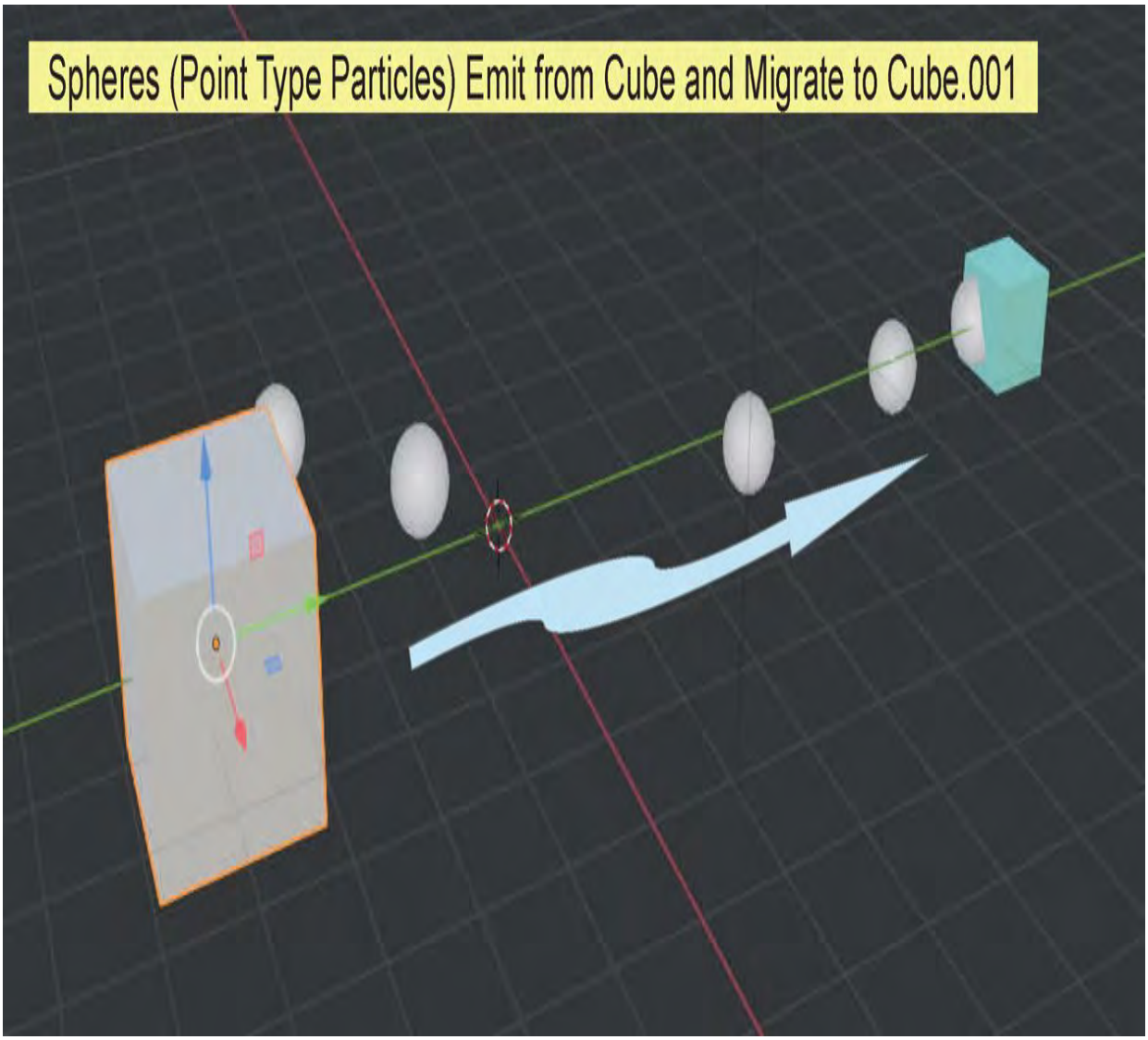
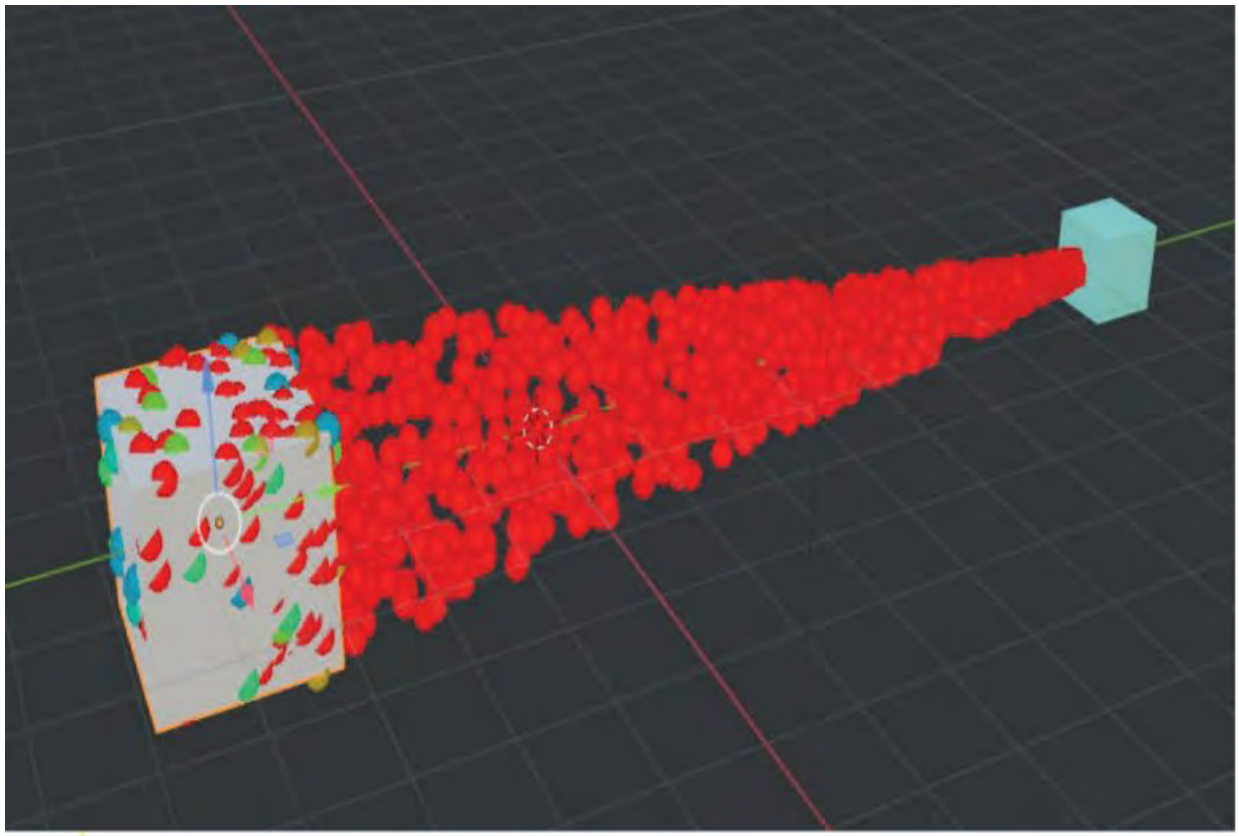


Figure 24.64

Remember that the location of either or both of the Cubes in the Scene may be animated at the same time. Animating the target Cube can cause the Particles confusion. They may head over to where the target Cube was originally located , have a think, then chase the target. Some Particles may take off in a completely different direction but in letting the animation play on

they will eventually find out they have made a mistake and discover where they should be going.

Emission Number 3000, Viewport Display – Color Velocity and Acceleratior



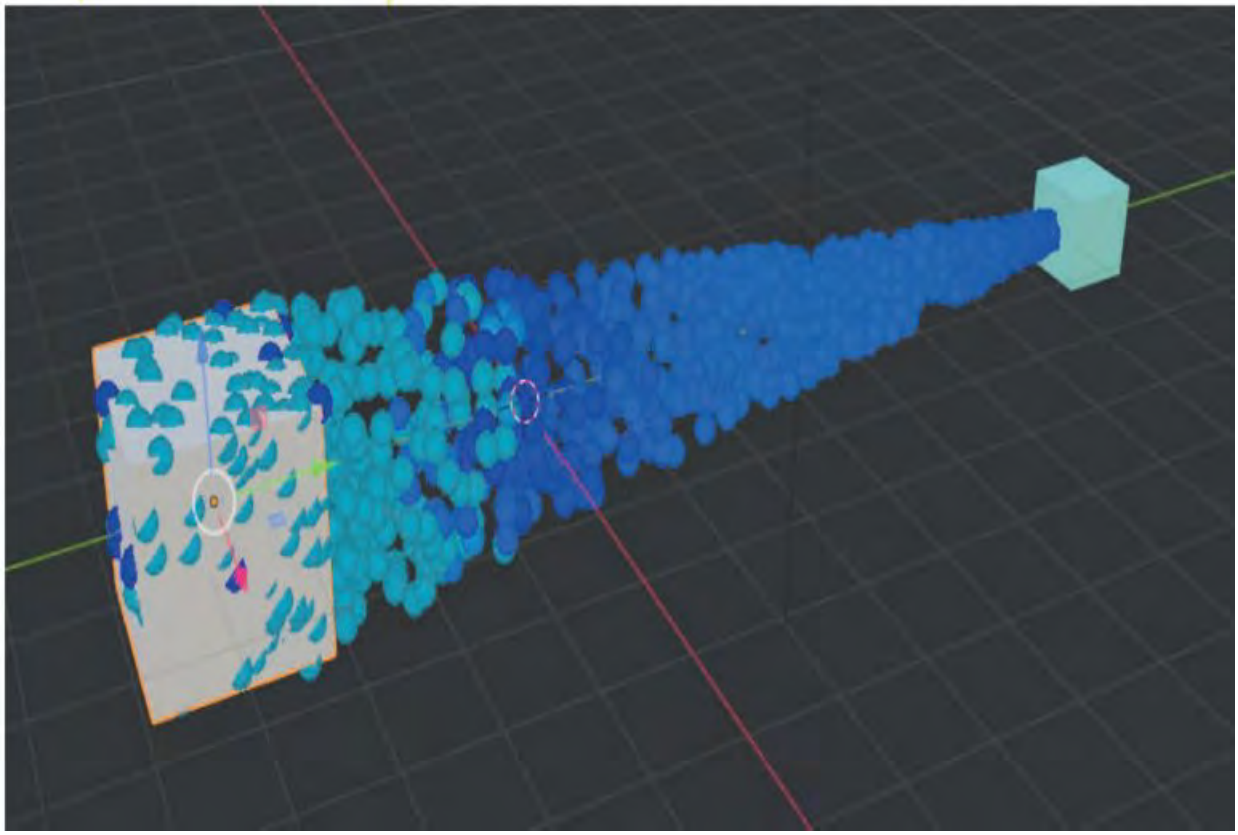


Figure 24.65

24.16 Hair Particles

Hair Particles are rendered as strands and may be edited in the 3D Viewport Editor. They are used to represent such things as grass, fur, hair, or anything that has a surface with fibrous strands.

In the 3D Viewport Editor, delete the default Cube object, add a Plane, and zoom in.

With the Plane selected, go to the **Properties Editor, Particles Properties** and add a **Particle System**.

In the top panel of the Particles Editor select **Hair** (Figure 24.66).

The Plane in the 3D Viewport Editor will show long strands sticking up from the surface. The **Hair Length** value in the **Emission Tab** allow you to adjust the length of the strands. (Figure 24.66).

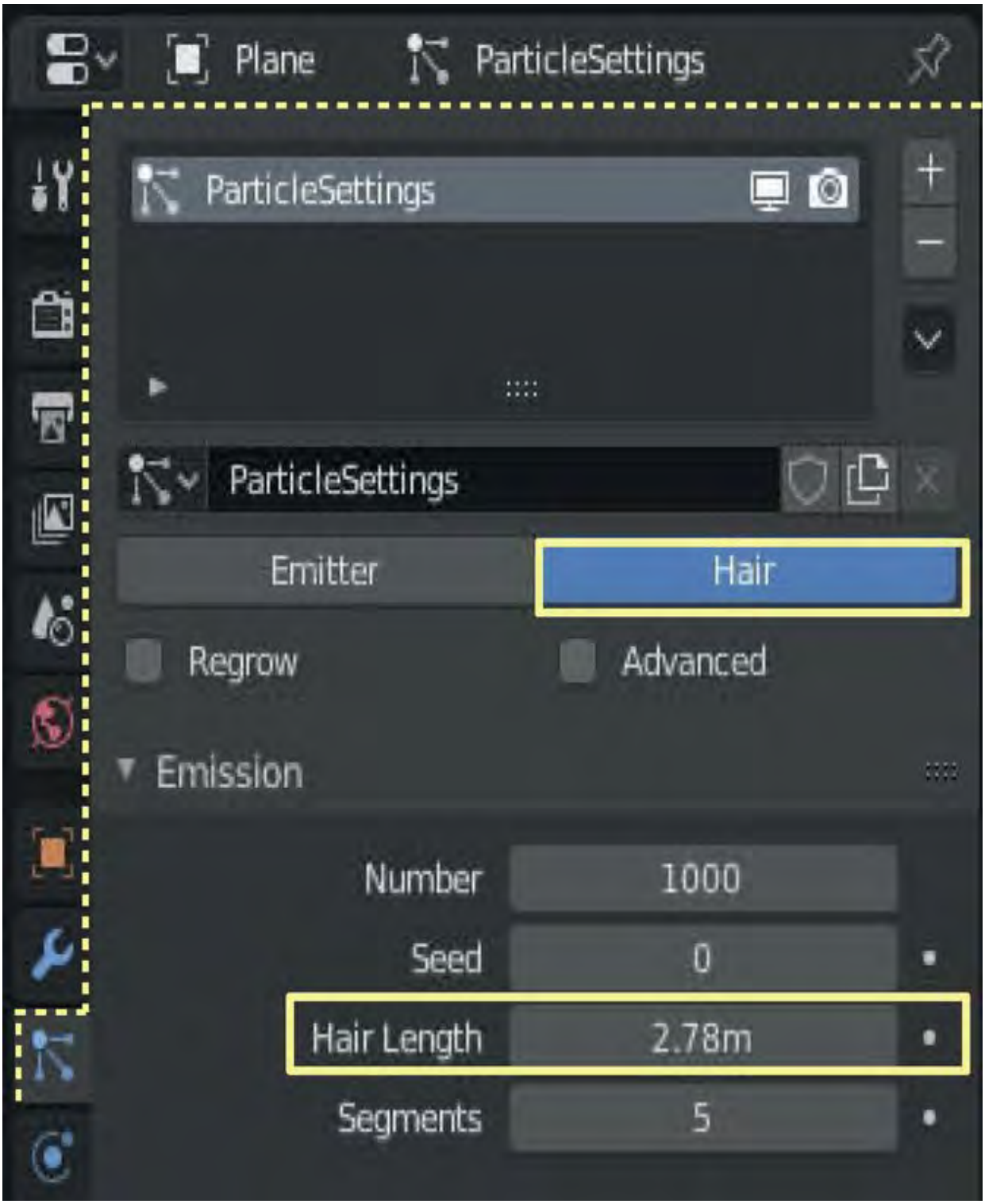
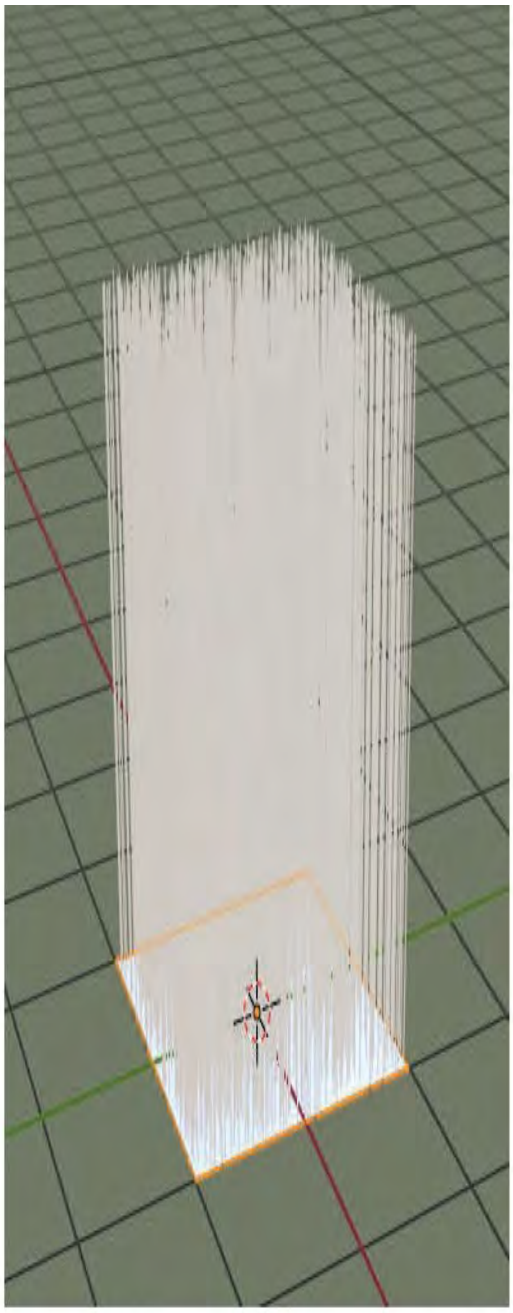
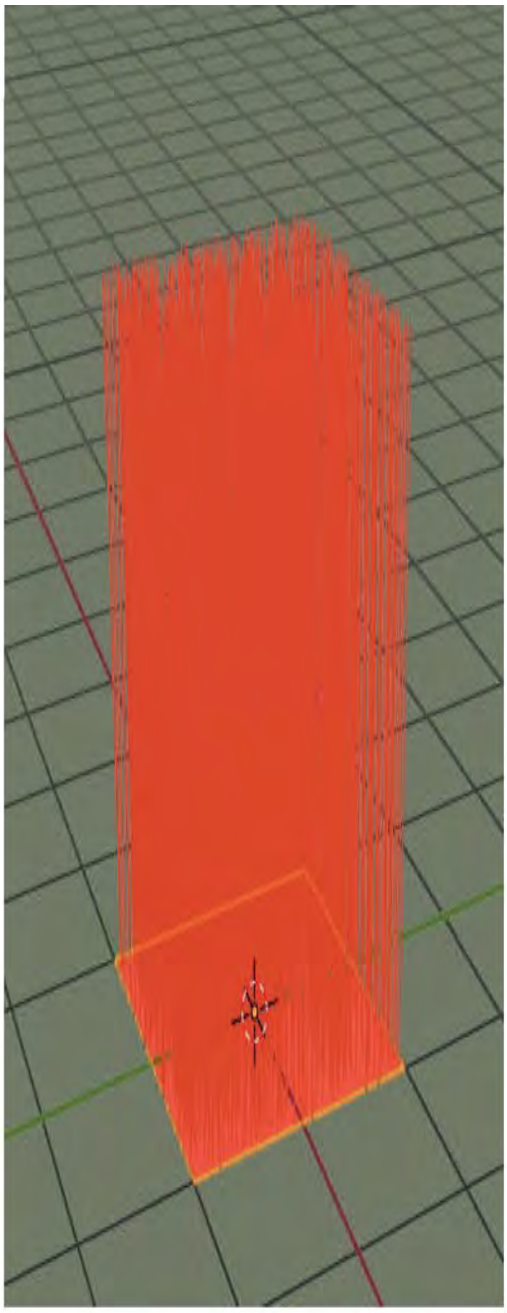

Plane
ParticleSettings
ParticleSettings
ParticleSettings
Emitter
Hair
Regrow
Advanced
Emission
Number
1000
Seed
0
Hair Length
2.78m
Segments
5

Figure 24.66



Hair Particles Emitted From Plane
Default Material
Hair Length: 2.64m



Hair Particles Emitted From Plane
Material Added to the Plane
Material Preview Viewport Shading

Figure 24.67

Hair Particles Emitted From Plane Default Material Hair Length: 2.64m Hair Particles Emitted From Plane Material Added to the Plane **Material Preview Viewport Shading**

Note: Hair Particles Render in the 3D Viewport Editor and take on the color of the Emitter Object.

Adding a Hair Particle System to a Plain Object would be suitable for generating a surface with fibrous strands such as a door mat or a patch of grass but a more interesting affect is to add Hair to a characters head.

Adding Hair to a Character

To demonstrate Hair will be added to a monkeys head (Suzanne the Monkey Mesh Object). In a new Blender file delete the default Cube an add a Monkey Object (Suzanne).

With Suzanne selected in the 3D Viewport Editor add a **Hair Particle System**.

Adding a **Hair Particle System** to **Suzanne** creates a hairy-headed Monkey with hair sticking out in every direction (Figure 24.68).

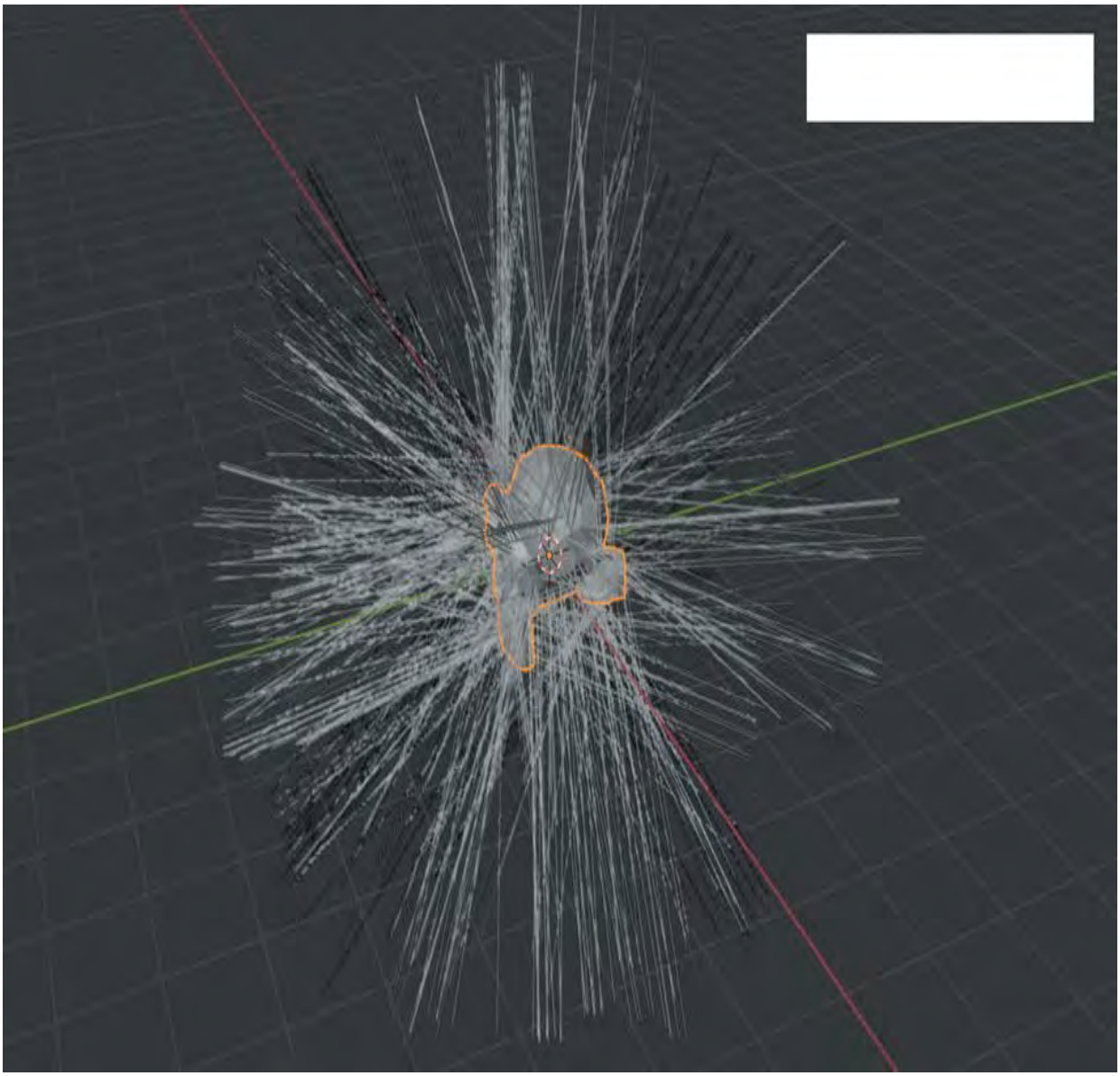


Figure 24.68

At this early stage make note that the Particle System is named **ParticleSystem** with settings named **ParticleSettings** linked (Figure 24.69). This is unimportant just now but will have significance as you progress.

Try for a more clean-cut look. **Remove the Particle System** by clicking the minus button (Figure 24.69)

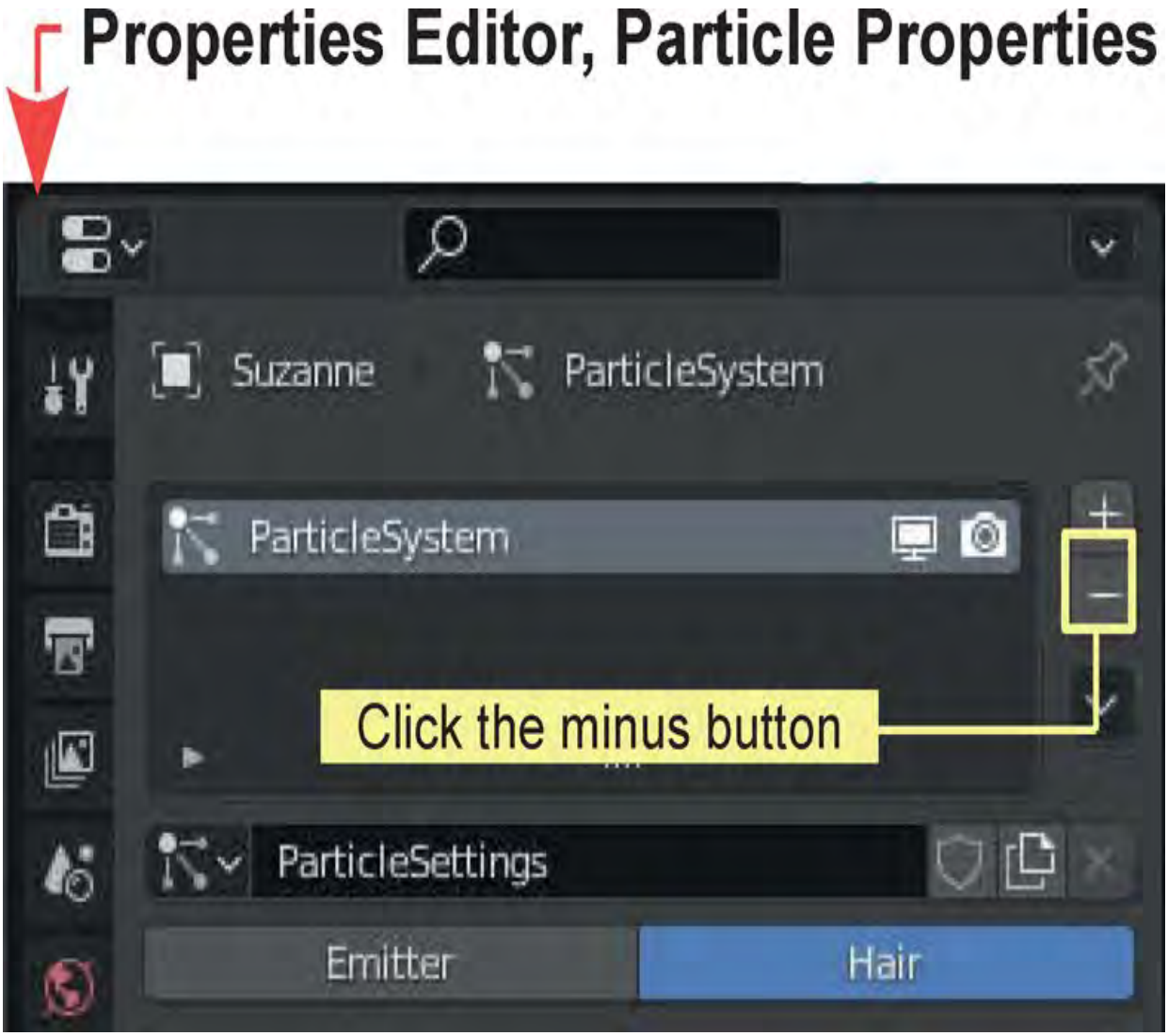


Figure 24.69

You designate specifically where the hair is to grow by selecting a **Vertex Group** (Chapter 05 – 5.9 Creating Vertex Groups).

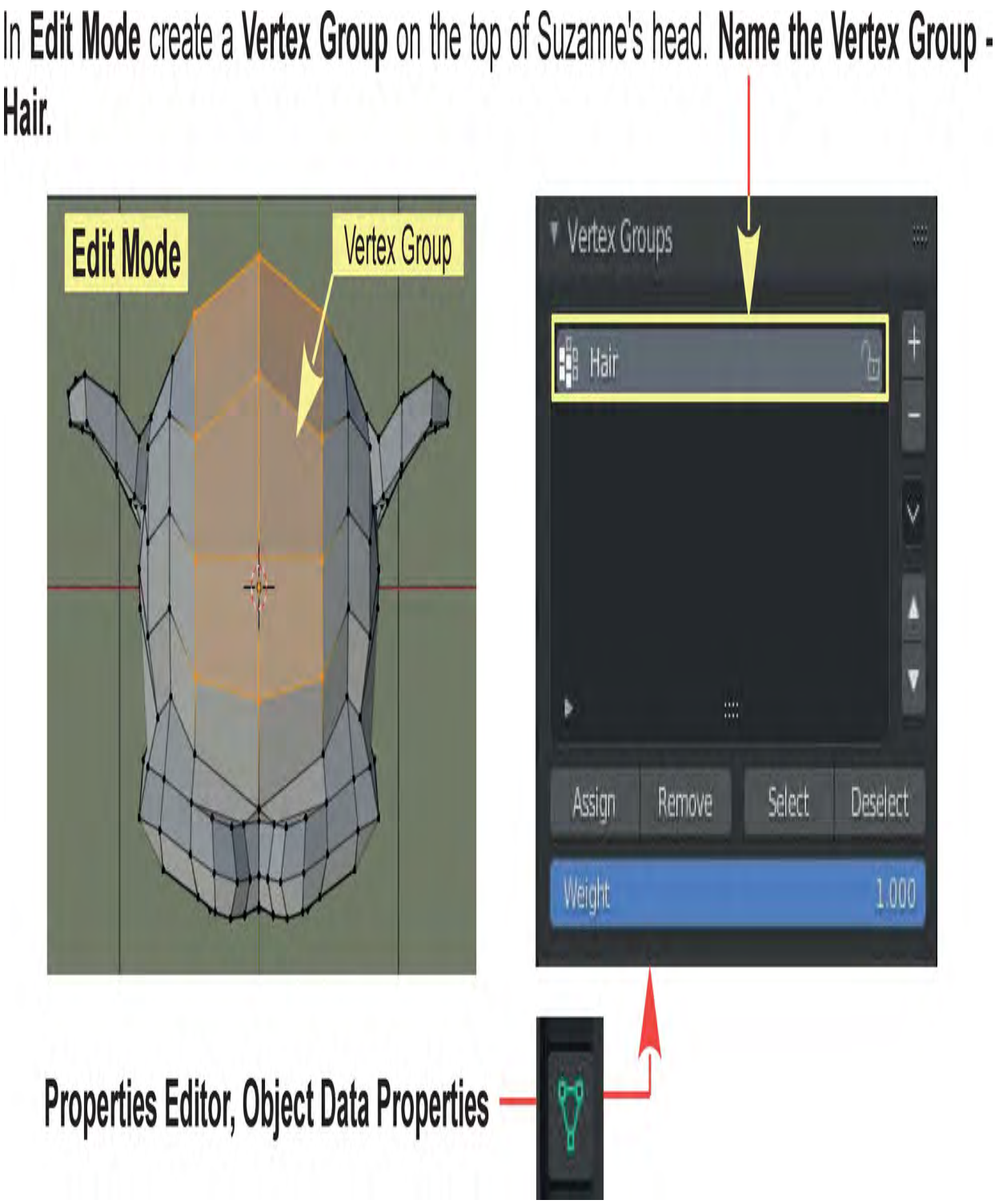


Figure 24.70

At this point you have nominated an area on the head by selecting a group of Vertices. **You do not have hair. While remaining in Edit Mode** go to the

Particles Properties and add a new **Particle System**. Change **Type: Emitter** to **Type: Hair**. Nothing happens because you are in **Edit mode**.

Note: Blender has named the new Particle System, **ParticleSystem** as before, but since you are working in the same Blender file, new Particle Settings named are **ParticleSettings.001** are linked (Figure 24.71).

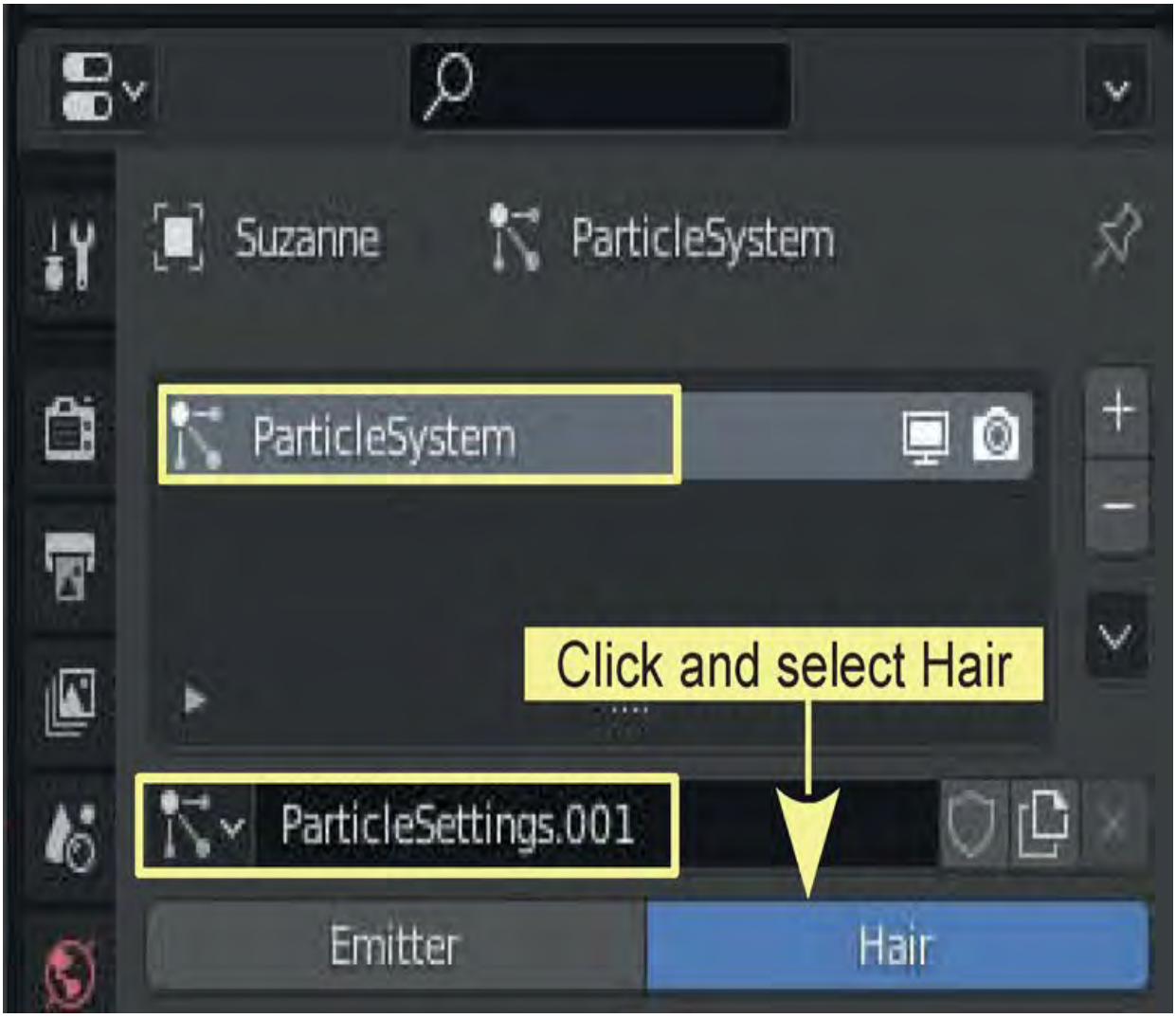


Figure 24.71

Tab to **Object Mode** in the 3D Viewport Editor and you will see plenty of hair. In fact, there is hair everywhere.

To correct this look, in the **Hair Length box** in the **Emission Tab,** decrease the value until the hair strands look reasonable; say, about 0.820.

Remain in the Particles Properties, go down to the **Vertex Groups Tab** and in the panel next to **Density** click and select **Hair** from the menu that displays (Figure 24.72).

Figure 24.72

Hair only displays on the area selected (the Vertex Group). Press Num Pad 3 to get a side view (Figure 24.73). To fix the scrawny look, go to the

Children Tab in the **Particle Properties** and click on **Simple** to get a bushy, Mohawk (Figure 24.74).

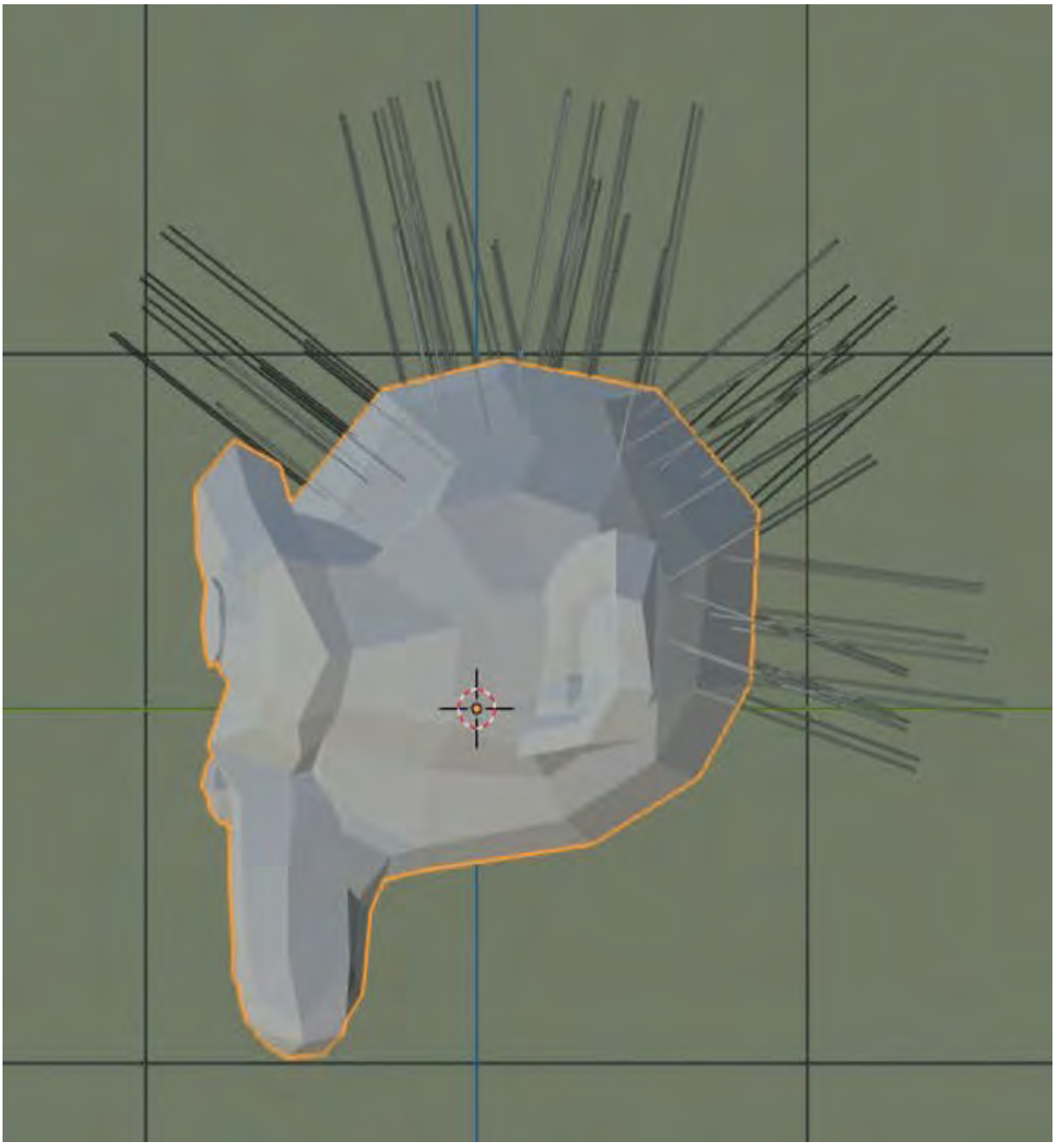


Figure 24.73

Figure 24.74

Suzanne is beginning to look like some cool dude in a cartoon feature but needs a little more character. A beard could do the trick.

Adding a Beard

To make the process interesting, vary the procedure just a little. You previously created a Vertex Group by selecting Vertices in Edit Mode and assigning the selected Vertices to the Group.

With **Suzanne** selected in **Object Mode** (with the new head of hair) in the 3D Viewport Editor, go to **Properties Editor, Object Data Properties** and click on the plus sign in the **Vertex Groups Tab** to add a new Vertex Group. Blender names the new group, **Group**. Rename this to **Beard** as you did before for the hair. You now select Vertices to assign to the new group. You could use the procedures as outlined previously, but do it a different way.

Tab into **Edit Mode** and **deselect** all the Vertices. In the 3D Viewport Editor Header, change to **Weight Paint Mode**. Suzanne turns blue in the 3D Viewport Editor, which indicates that no Vertices are selected (Figure 24.75).

Figure 24.75

Note that **Beard** is highlighted in the **Vertex Groups Tab**. Click on **Hair**. If you select (highlight) the Hair Vertex Group and look closely amongst all that hair (it may help to rotate the view), you will see a red scalp; this is showing the area that was previously selected. Tab to **Edit Mode** and click on **Select**

in the **Vertex Groups Tab** and you'll see the vertices that were painted for Hair. **Click LMB** in the 3D Viewport Editor to **deselect Vertices,** and click on Vertex Group **Beard** again in the **Object Data Properties, Vertex Groups Tab**. Select **Weight Paint Mode** in the 3D Viewport Editor and look at the **Tools Panel** at the left hand side of the screen (Figure 24.76).

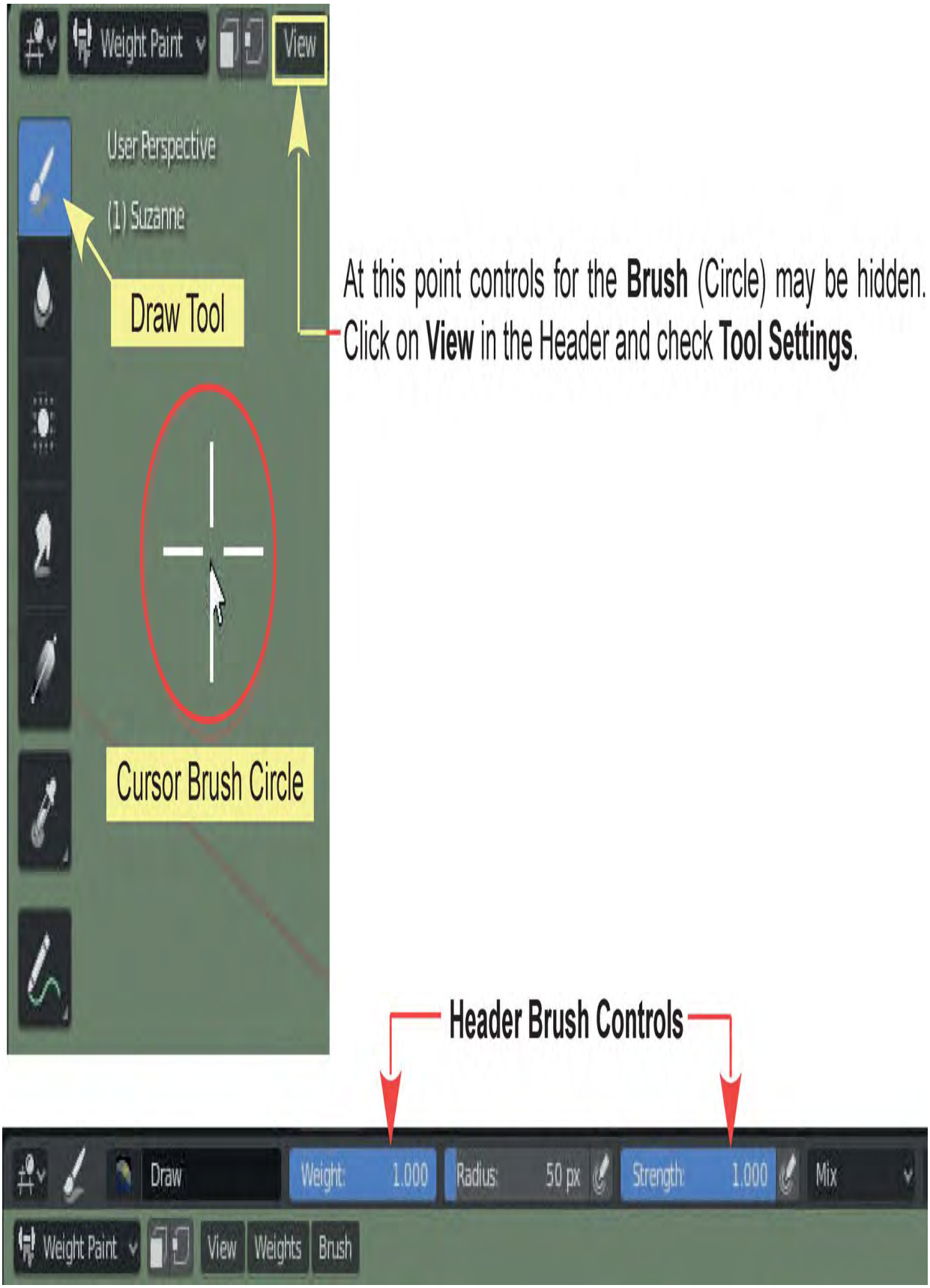

Weight Paint
View
User Perspective
(1) Suzanne
Draw Tool
At this point controls for the **Brush** (Circle) may be hidden.
Click on **View** in the Header and check **Tool Settings**.
Cursor Brush Circle
Header Brush Controls
Draw
Weight: 1.000
Radius: 50 px
Strength: 1.000
Mix
Weight Paint
View
Weights
Brush

Figure 24.76

By default the **Draw Tool** is selected and the Cursor is a circle.

Controls for the Brush display in a new Header. **Note;** The **Weight and Strength** values in the Header are 1.000. In the 3D Viewport Editor, click LMB, hold and drag the Cursor (circle) over Suzanne's chin. You will see the color change as you drag. Rotate the view to paint the area where you want Hair.

Keep dragging until the chin is all red (rotate the 3D Viewport Editor as you paint). The red chin means that you have selected this area as the new Vertex Group for the beard (Figure 24.77).

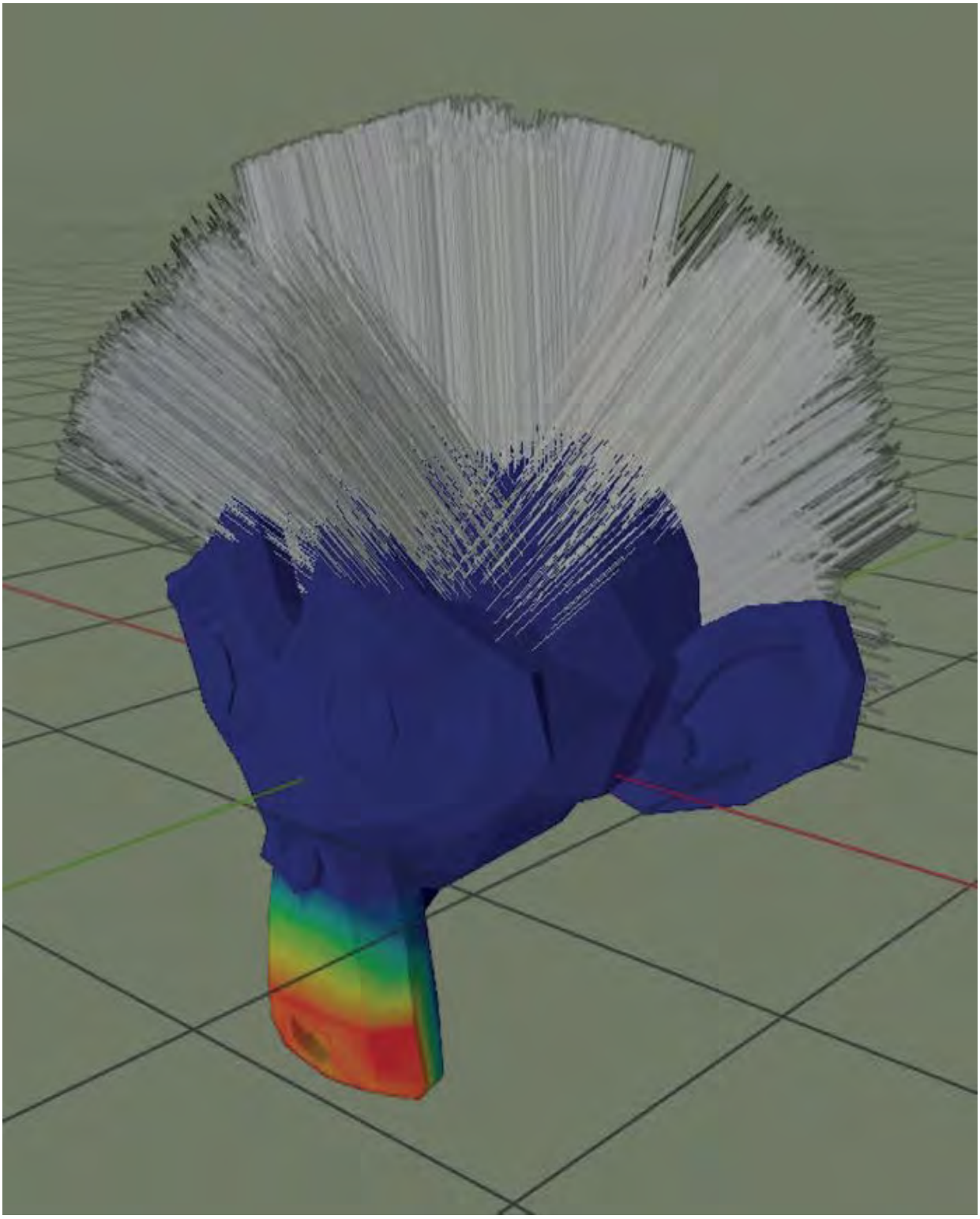


Figure 24.77

Tab to **Edit mode,** making sure all Vertices are **deselected,** then in the **Vertex Groups Tab,** with **Beard** highlighted (selected), click on **Select** (Beard Vertices Display) then click **Assign** to assign the painted Vertices to the beard Vertex Group. Click **Select** to see them.

With the painted Vertices selected (you are in Edit Mode) go to the **Particles Properties** and add a second **Particle System** (click on the plus sign next to where you see **ParticleSystem** highlighted).

Note: that Blender names this new system **Particle System.001** with **ParticleSettings.002** linked. ParticleSystem.001 is a channel which assigns the information to create the Particles for the Vertex Group. PaticleSettings.002 is the Datablock containing the information.

In creating new Particle Systems in a Blender file, the automatic process generates new Datablocks and stores them for future use. You see the **Datablocks** by clicking the **Browse Particles Settings to be linked** button.

Figure 24.78

Select **Type: Hair,** decrease the **Hair Length value** in the **Emission Tab** to 0.290, and go down to the **Vertex Groups Tab,** click in the **Density box,** and select **Beard**.

In the **Render Tab** check that **Path** is in the **Render As** panel. **Tab** to **Weigh Paint** Mode in the 3D Viewport Editor. You will see the WeightPainted chin with a bunch of scrawny hair on the Suzanne's chin. Go to the **Children tab** and click **Simple** for a hairy Beard (Figure 24.79).

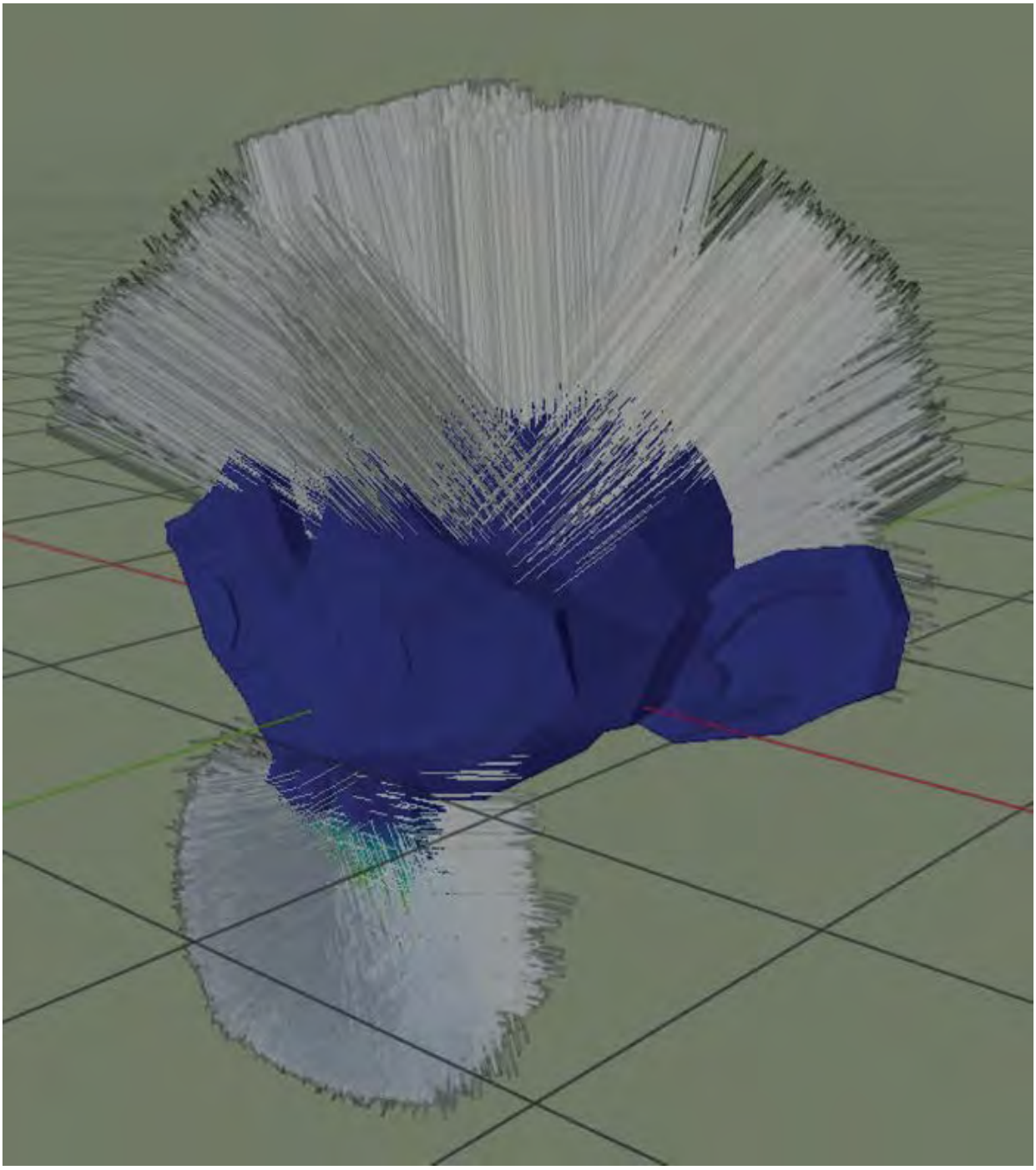


Figure 24.79

It doesn’t matter in which order you do it, the procedure is the same: select Vertices to define the area, create a Vertex Group, assign Vertices to the Vertex Group, create a Hair Particle System and assign it to the Vertex Group.

Note: If you elect to select Vertices, assign them to a Vertex Group then add a Hair Particle System you may find the Hair displaying as dots instead of strands. **In the Render tab click on Path**.

In Object Mode in the 3D Viewport Editor you have a gray Monkey with a grey beard and hair (Figure 22.70). This is fine, but it isn't all that exciting in a render. Jazz it up by adding different **Materials** to Suzanne's surface creating **Hair Colors** for the hair and beard.

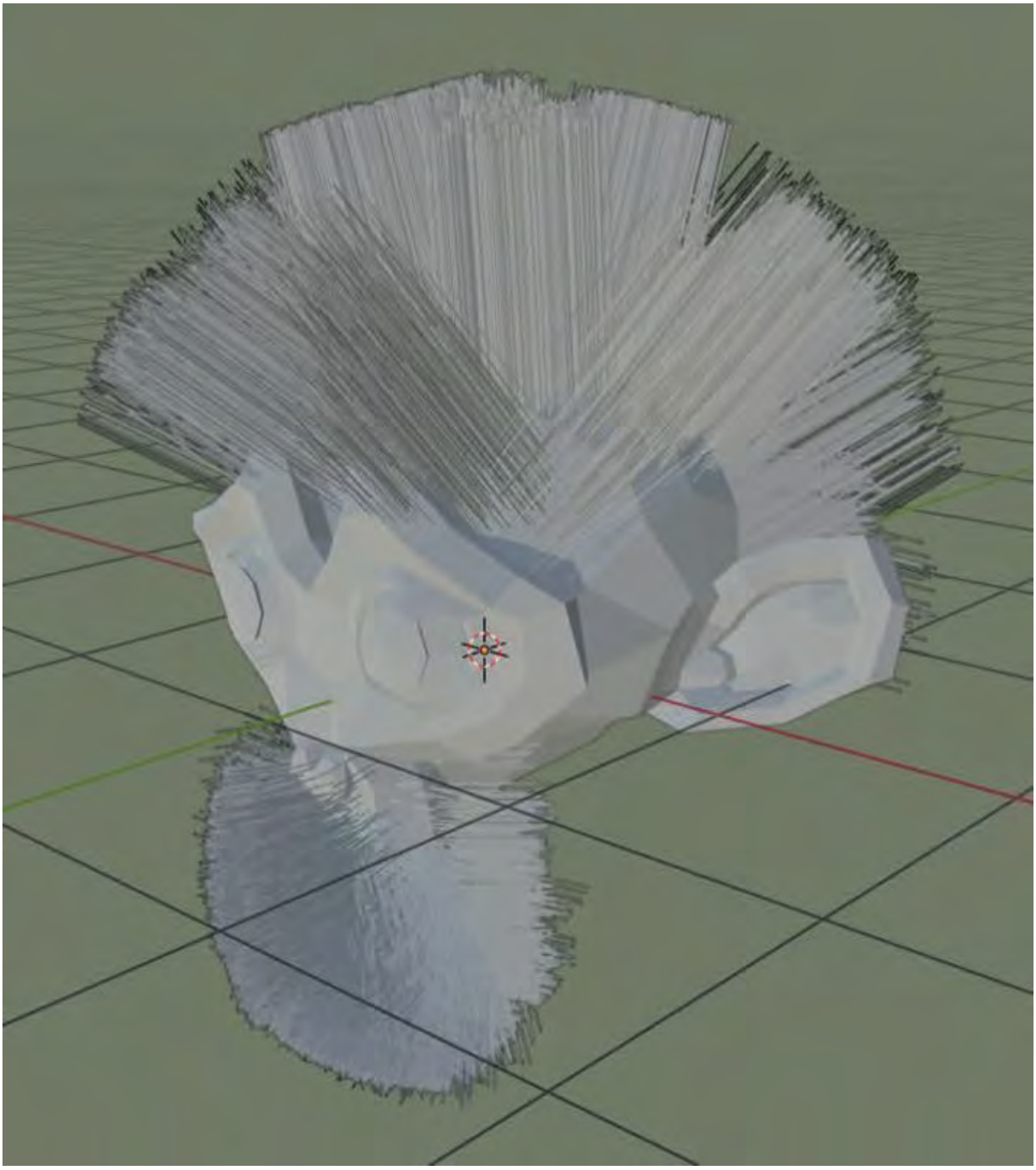


Figure 24.80

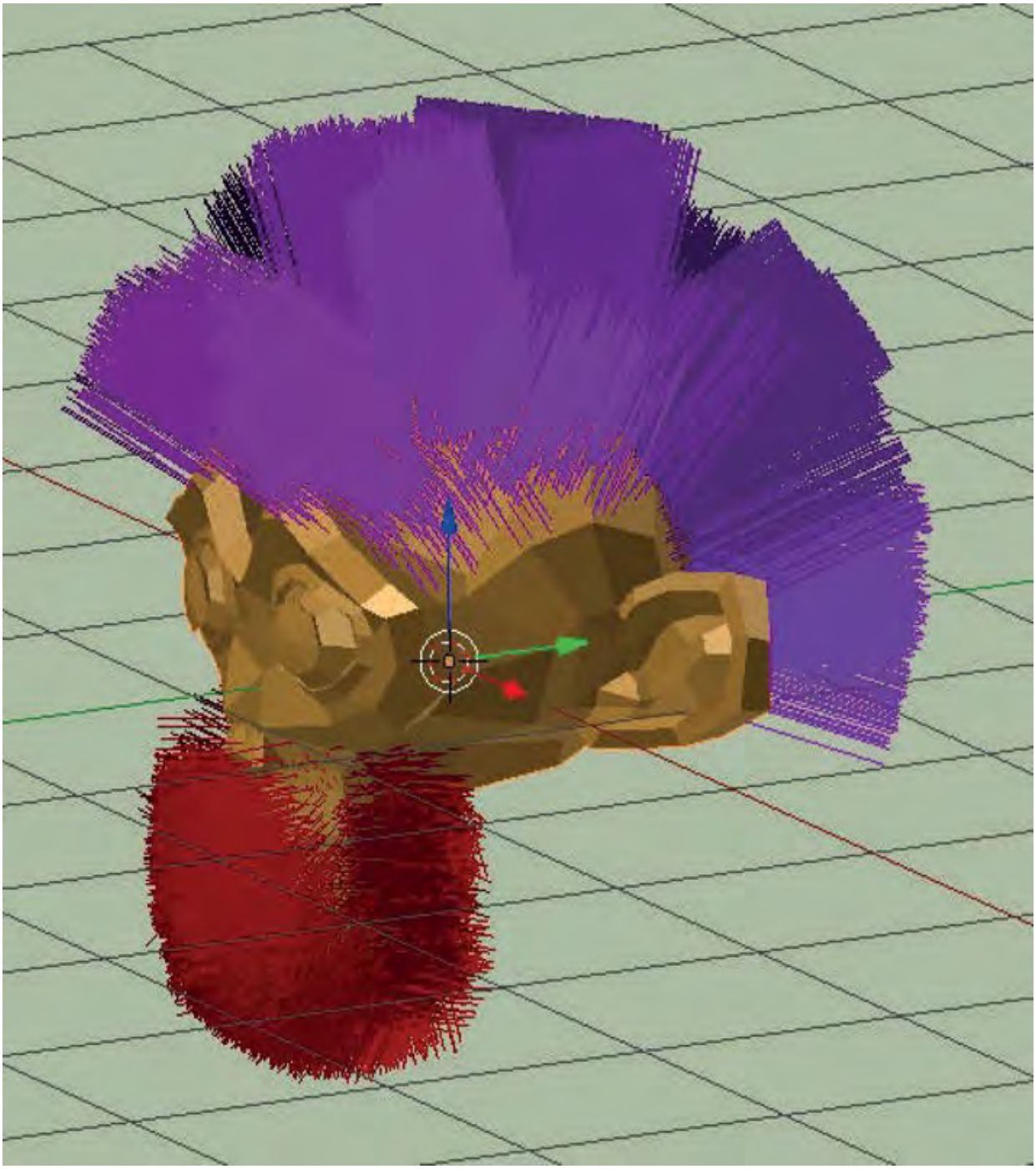


Figure 24.81

Reference Two Vertex Groups (Chapter 16– 16.7).

In Object Mode Suzanne with Hair displays with the default gray color (Figure 24.80). The Hair Particles (Strands) take on the Material color of the

Emitter which, in the case of this demonstration, are the Vertex Groups that have been created.

To color the Hair you create Material Slots (colors) and assign the slots to the Vertex Groups (Reference Chapter 16 -16.7).

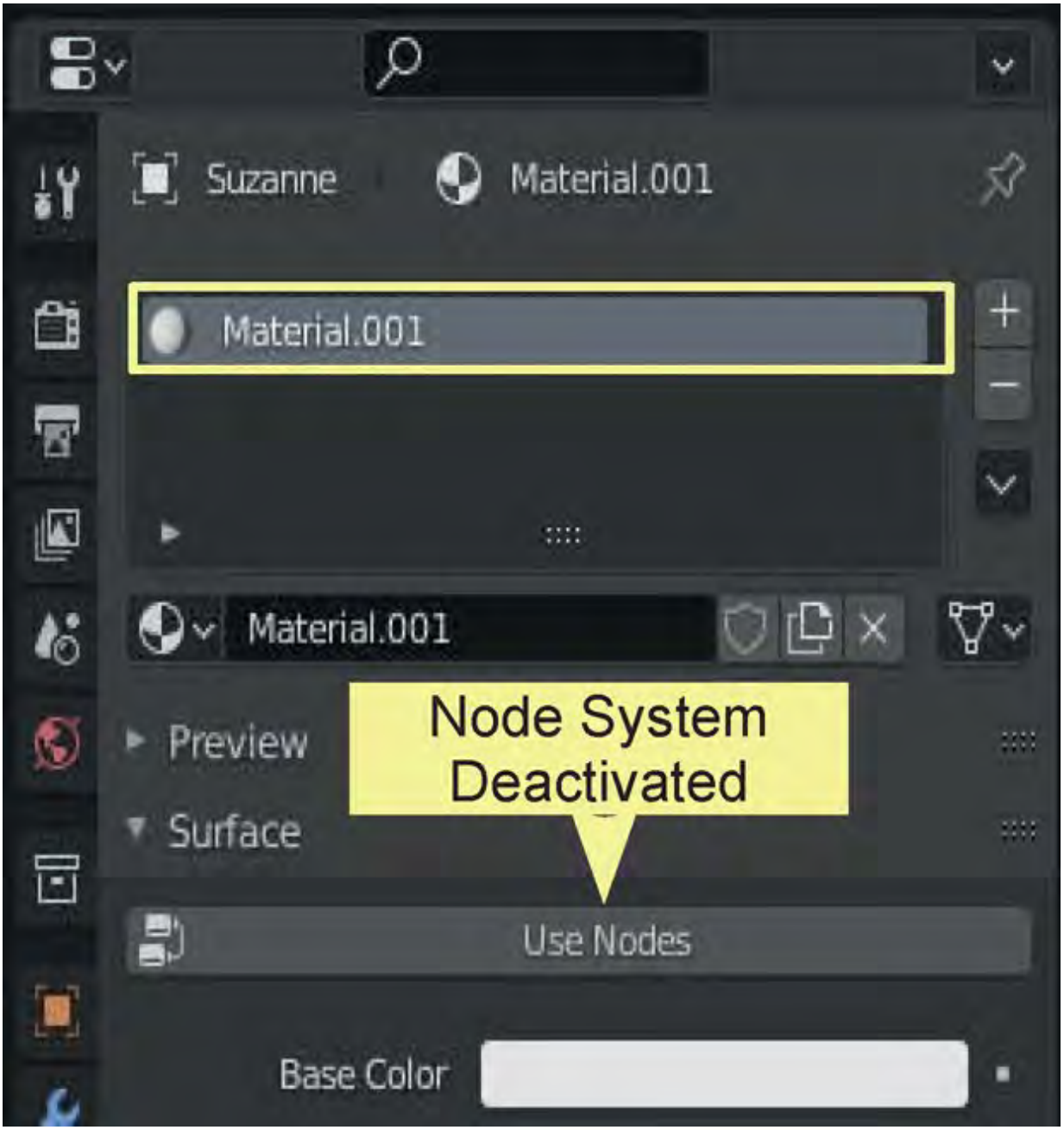


Figure 24.82

Have Suzanne selected in the 3D Viewport Editor. If you haven't applied a Material to Suzanne she will display with the default gray color. In the Properties Editor, Material Properties only the New Button will display. Click on New to generate a Material Slot. The Slot is named Materia1.001. Click Use Nodes to cancel Blenders Node System (Figure 24.82).

Note: This procedure is described with the Blender Node System deactivated to allow Materials (colors) to display in the 3D Viewport Editor in Solid Viewport Shading Mode.

In the Properties Editor, click the Plus sign adjacent to the Materia1.001 Slot. To create a new Slot click New and cancel the Node The new Slot is named Materia1.002.

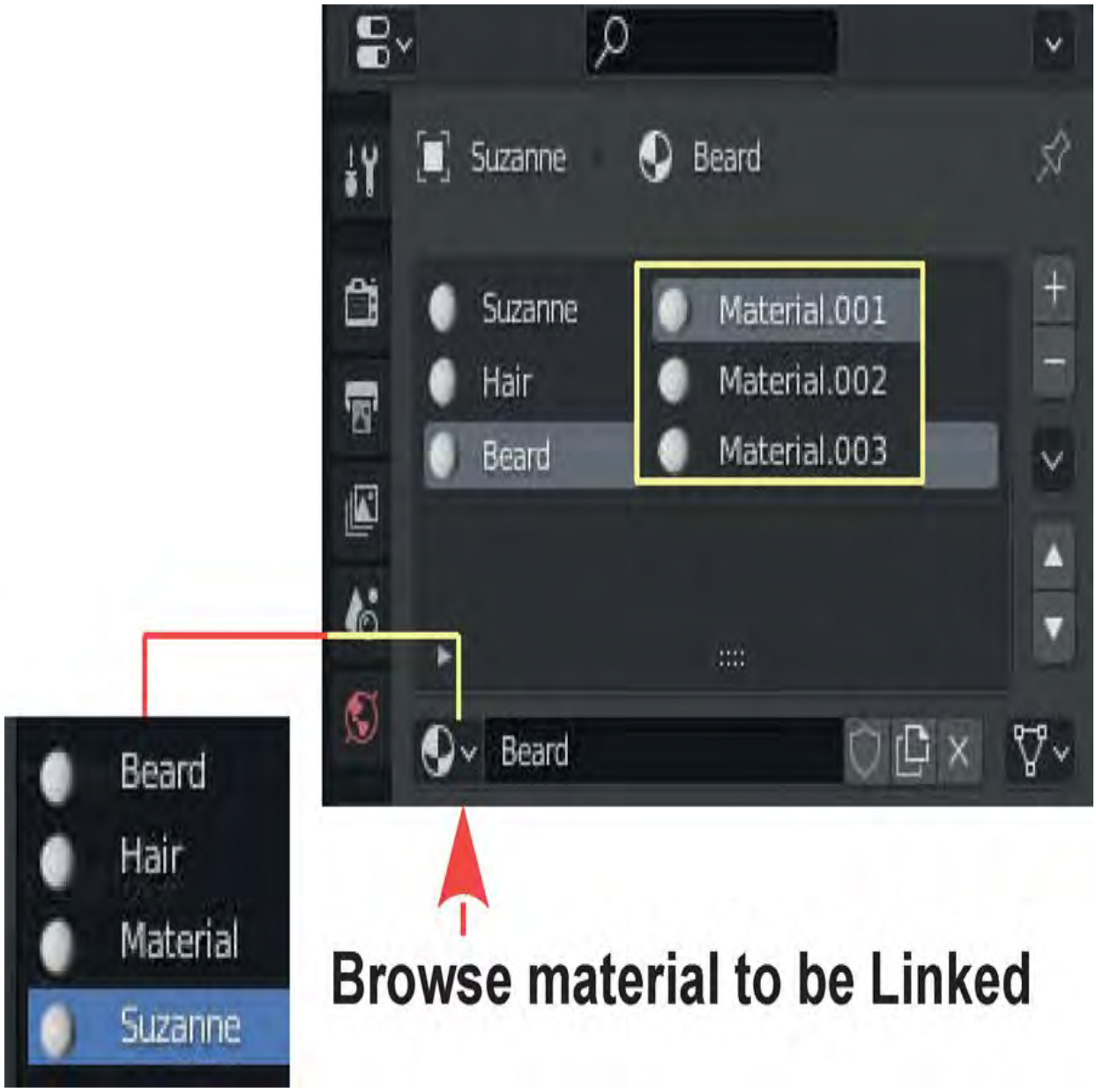


Figure 24.83

Repeat the procedure to create a third Slot (Materia1.003).

Why the three Slots? You have two Vertex and Suzanne herself to color.

At this point the three Slots all have the default gray Material assigned. Before assigning different colors rename the material Slots. Make Materia1.001, **Suzanne**. Materia1.002, **Hair** and Materia1.003, **Beard**.

With the Slots renamed, if you click on each in turn you will see the names change in the Browse Material to be Linked button. Opening the Cache show the same Material names plus the default Material named **Material**.

Click on the **Hair Slot** (cancel the Node System) and choose a purple Material color for the hair. Click on the **Beard Slot** and choose a red color for the beard.

Suzanne does not change color in the 3D Viewport Editor. She has the Material named Suzanne from the Cache (default gray) assigned to the Slot name Suzanne which is in turn applied to Suzanne's Mesh surface in the 3D Viewport Editor.

To apply color to the Hair have Suzanne in Edit Mode with all Vertices **deselected**.

In the **Properties Editor, Object Data Properties,** click on Hair in the Vertex Groups Tab Click Select to show the Vertices in the Hair Vertex Group.

In the **Material Properties,** click on the material Slot name **Hair** (purple). Still in the Material Properties click **Assign** to assign the Hair material Slot to the Hair Vertex Group.

Go back to the Object data Properties and select (click on) the Beard Vertex Group to select the Vertex Group.

Back to the **Material Properties,** click on the **Beard Material Slot** and click **Assign** to assign the Slot to the Vertex Group.

The Materials are assigned to the Vertex Groups but in Object Mode everything is gray?

Figure 24.84

Go to the **Properties Editor, Particle Properties** and select **ParticleSystem** (the Particle System for the Hair). In the **Render Tab** (Figure 24.84), click the Material bar and select **Hair** (the material Slot for Hair).

Repeat for the **Beard** selecting **ParticleSystem.001** and **Material: Beard** (red).

You now have a purple haired red bearded Suzanne! (Figure 24.85). She is a scruffy looking character which could benefit from some tender grooming. This may be performed in **Particle Edit Mode**.

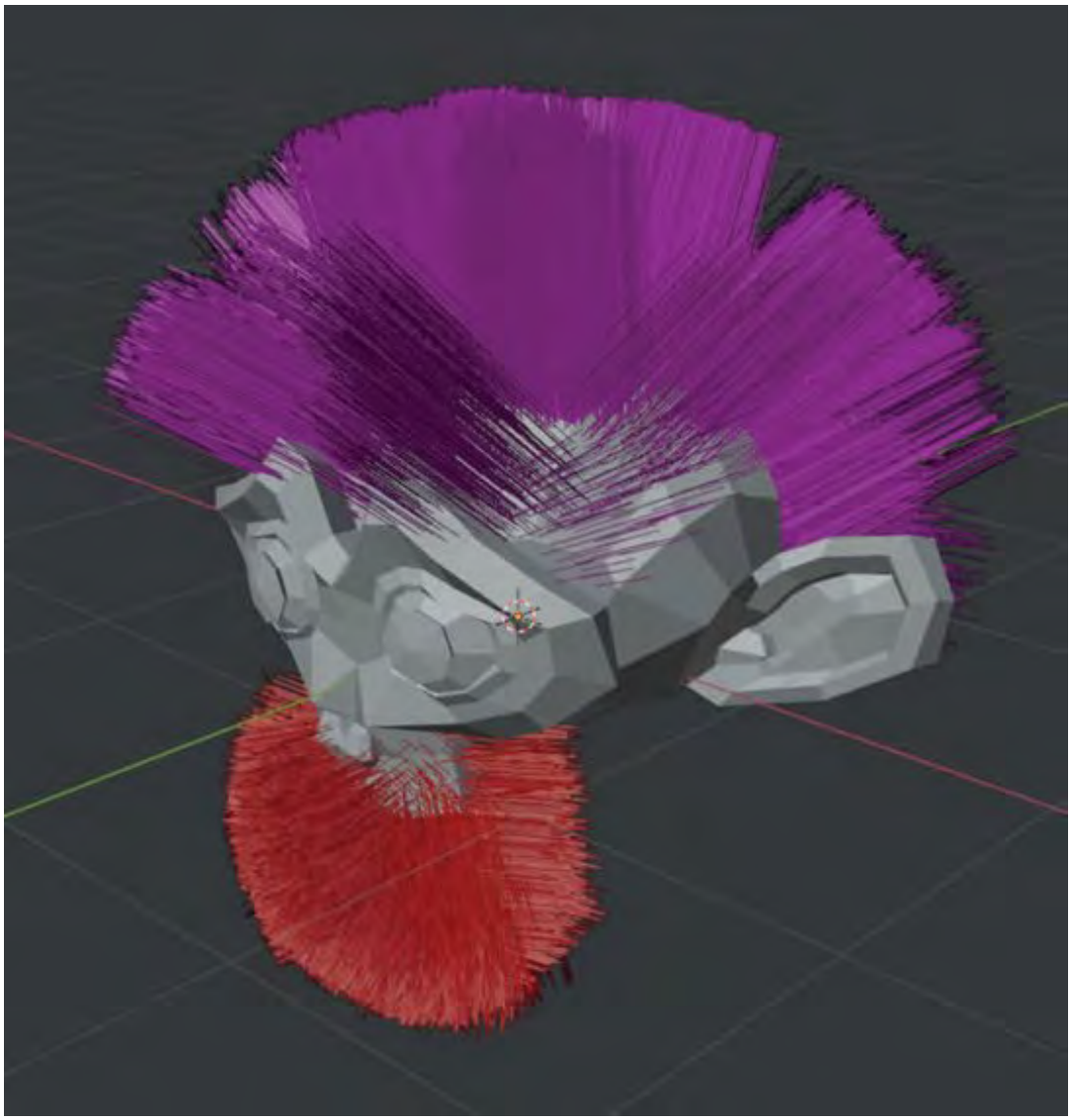


Figure 24.85

Note: Adding hair to an Object can add an awful lot of Vertices, which, when rendering, can take an awful lot of time and may even cause your computer to stall out. If you are not doing anything serious and have a slow machine to start with, keep the number of strands low.

In attempting to demonstrate how to groom the bearded Suzanne it was discovered that the computer was struggling, therefore, the technique will explained using a simple Plane Object with Hair Particles applied.

24.17 Particles Edit Mode

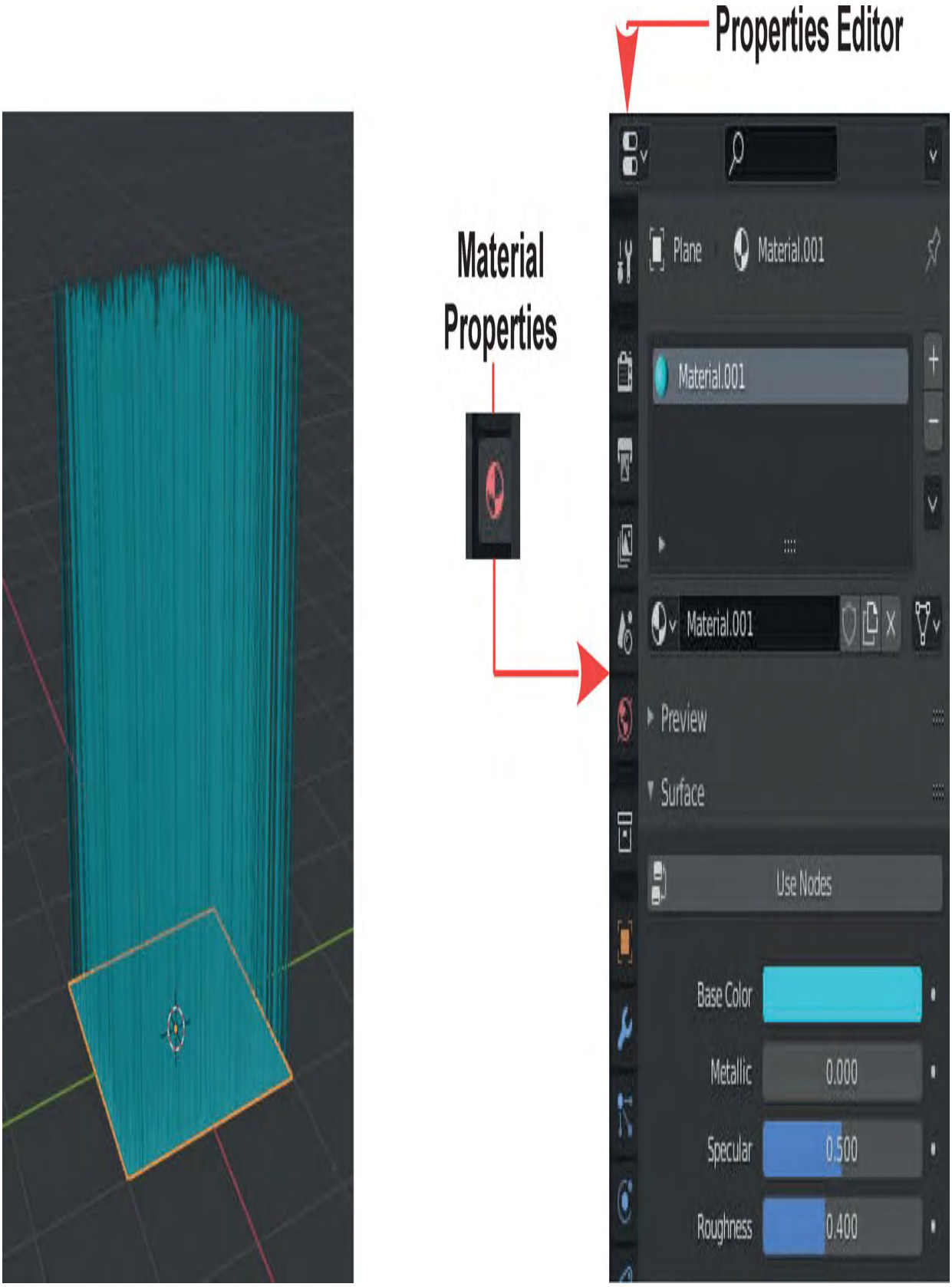

Properties Editor
Material
Properties
Plane
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Base Color
Metallic
0.000
Specular
0.500
Roughness
0.400

Figure 24.86

Figure 24.86 shows a Plane Object in the 3D Viewport Editor with a blue **Material** and a **Hair Particle System** applied.

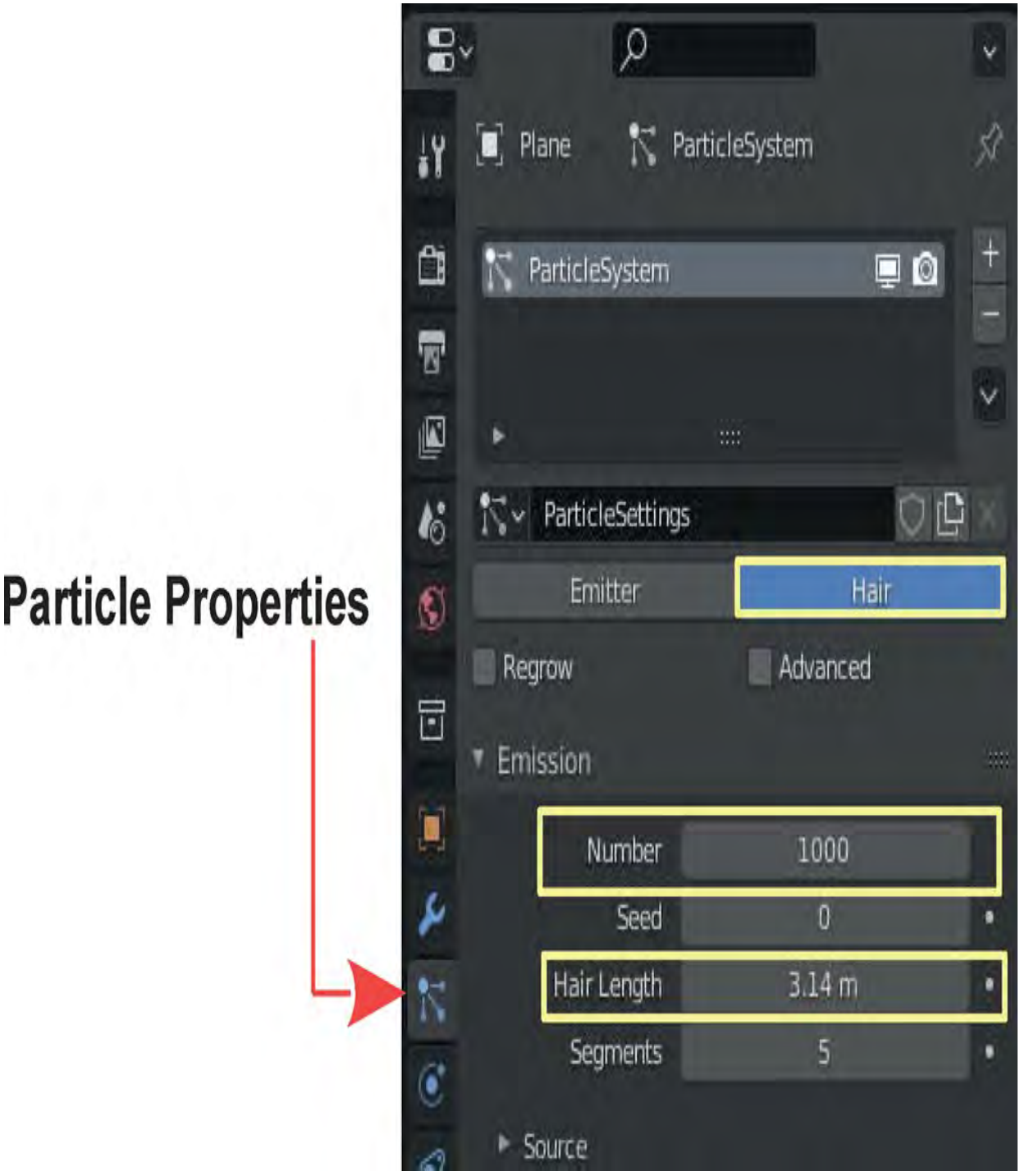


Figure 24.87

Figure 24.87 shows the settings for the **Hair Particle System** creating the display. With the Particle System applied switch to **Particle Edit** Mode in the Header.

Note: The **Particle Edit Mode** option only displays when a Hair Particle System has been applied to the selected Object in the 3D Viewport Editor.

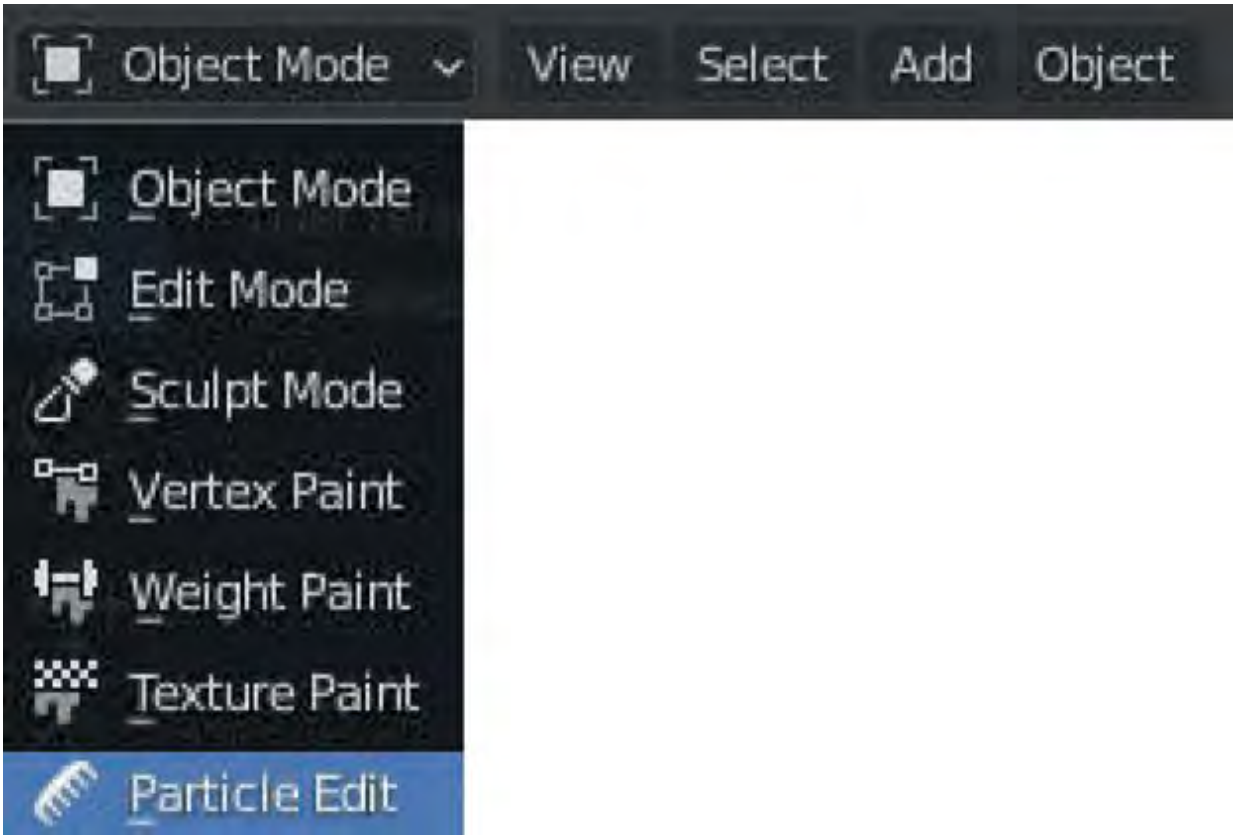


Figure 24.88

3D Viewport Editor - Particle Edit Mode

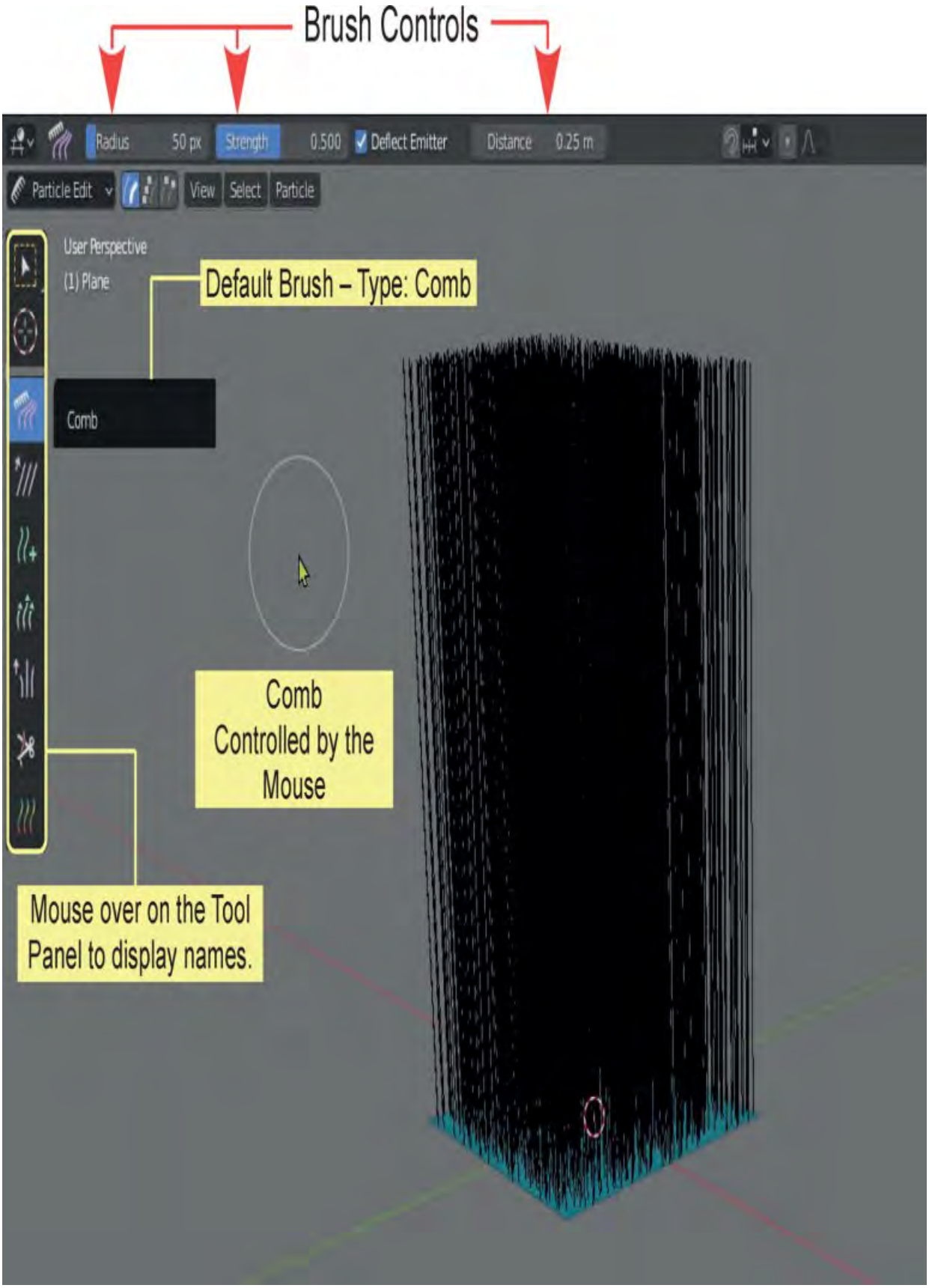
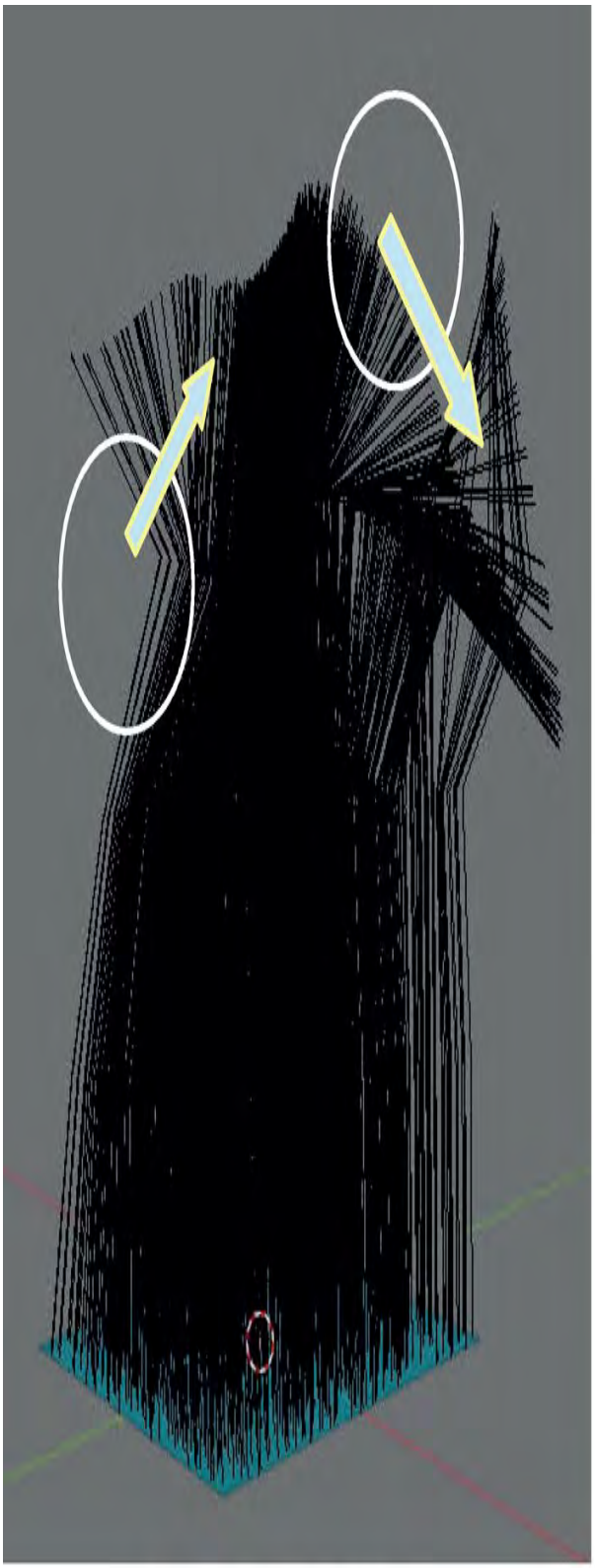

Brush Controls
Radius
50 px
Strength
0.500
Deflect Emitter
Distance
0.25 m
Particle Edit
View
Select
Particle
User Perspective
(1) Plane
Default Brush – Type: Comb
Comb
Comb Controlled by the Mouse
Mouse over on the Tool Panel to display names.

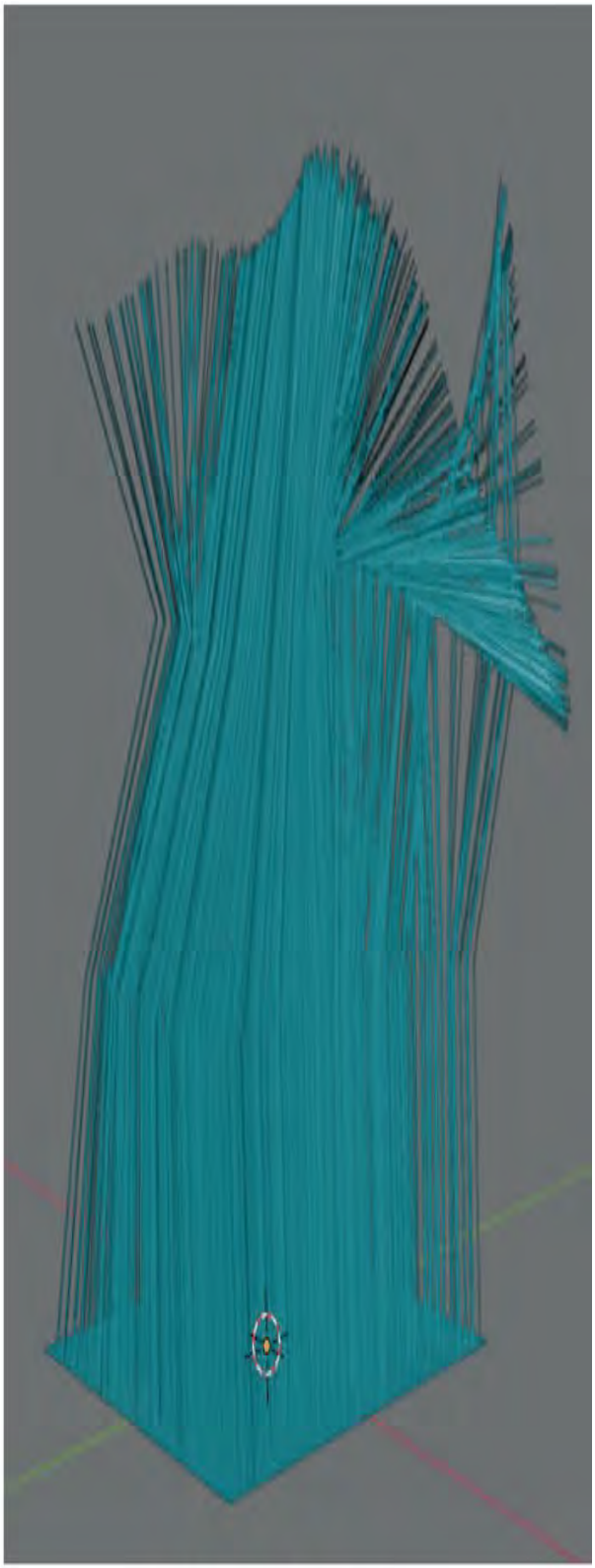
Figure 24.89

To Edit the Hair Particle display, stroke (comb) the Particle strands by dragging the Mouse.

Repeat strokes as required.



Particle Edit Mode



Object Mode

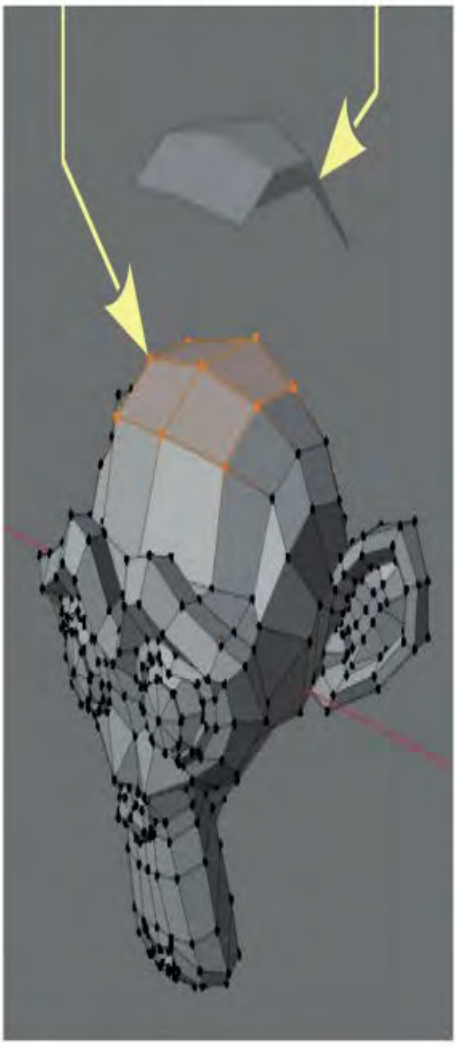
Figure 24.90

Experiment with the Brush Types and settings to become conversant with the operations.

As previously mentioned, adding Hair Particles can create a lot of Vertices, which when combined with the Vertices of the Mesh Model, may cause the computer to struggle in using the Particle Edit Mode. One technique to get around this dilemma is to duplicate only part of the Mesh Model to which Hair Particles are applied. In the following example a scull cap is formed for Suzanne.

In Edit Mode select the part of the Mesh Surface to which Hair Particles are to be applied. Duplicate the selected Vertices, click RMB in the 3D Viewport Editor and select Separate – Selection in the Vertex Context Menu. This creates a separate Object. Add a Hair Particle System to the new Object. When using Particle Edit Mode there is a significant reduction in the number of Vertices involved in the Edit operation. The new Object is positioned, in this case the scull cap, on Suzanne’s head.

Vertex selection for the **Scull Cap**



Hair Particles applied to the **Scull Cap**

Figure 24.91

With this technique the computer will find it easier to handle the computations required when making Scruffy Suzanne not so Scruffy.

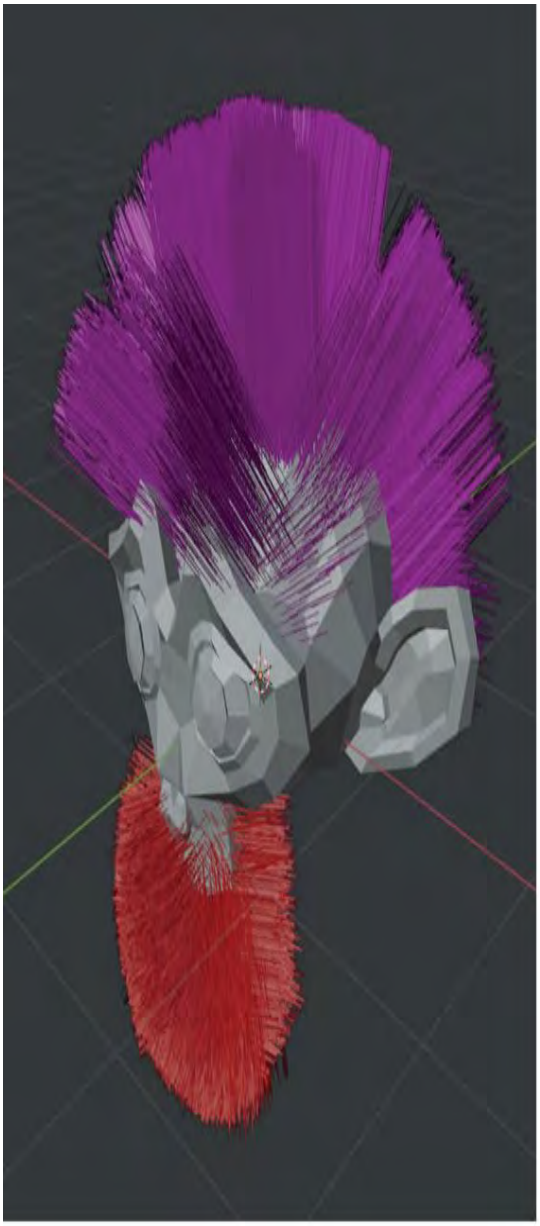


Figure 24.92

24.18 Particles for Arrays

Particles emitted from an animated Object may be used to create interesting Arrays such as a spiral (Figure 24.93).

To create the spiral, have a Plane Object in the Scene as the Emitter Object with an IcoSphere as the Rendered Object.

When you add the IcoSphere, the **Add Ico Sphere panel** displays in the lower LH corner of the Screen. Click on the panel and you will see Subdivisions: 2 as the default setting. Reduce this to Subdivisions: 1. This reduces the Vertex count on the surface of the Ico Sphere. This is not necessary here but is good practice when creating simulations where its advisable to use a minimum number of Vertices.

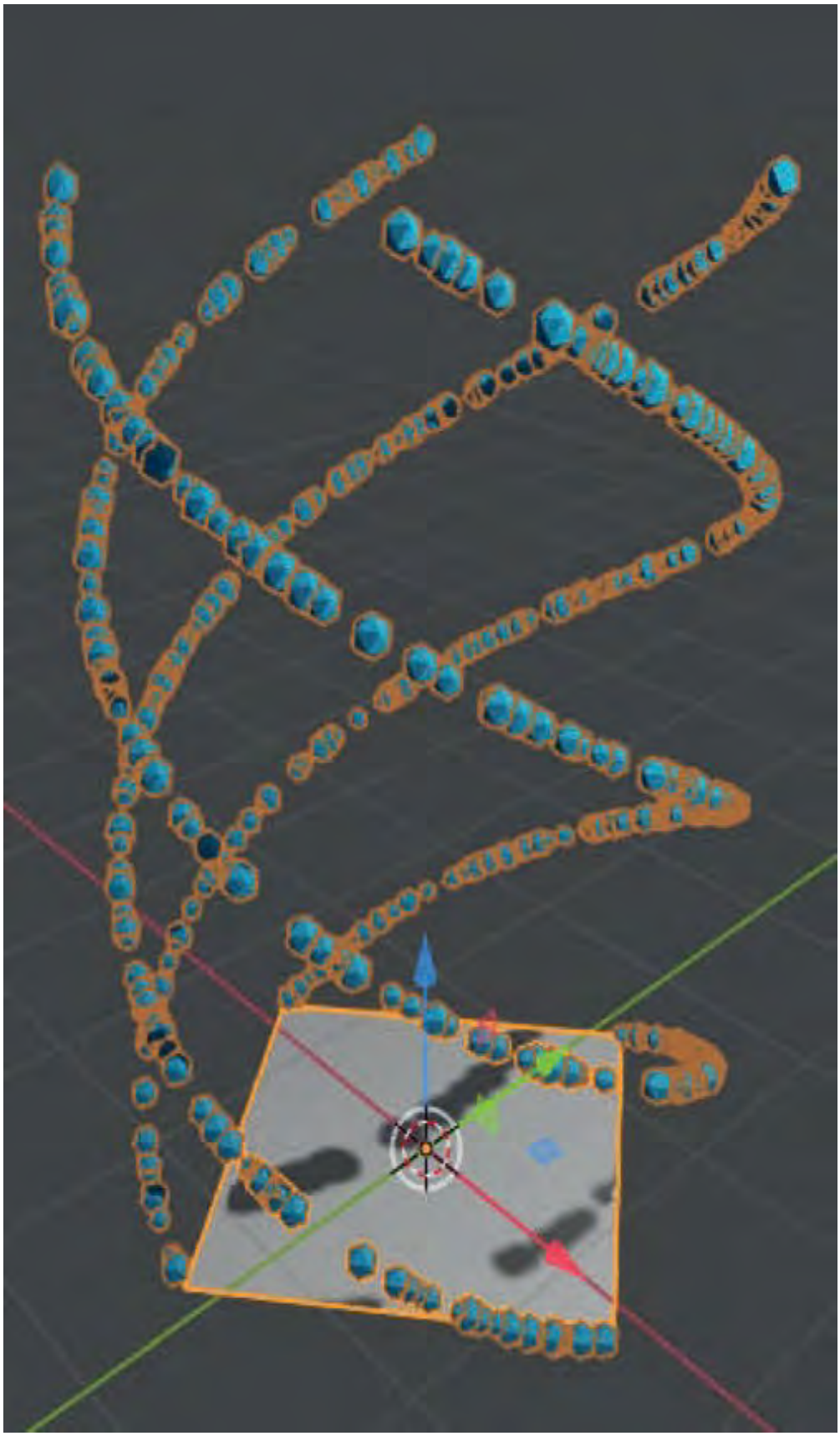


Figure 24.93

Park the IcoSphere off to the side of the Screen or you may place it in a separate Collection to hide it from view. Give the Ico Sphere a nice bright Material. Have the 3D Viewport Editor in **Material Preview Viewport Shading Mode**.

Animate the Plane to Rotate (See Chapter 20 – 20.24 Rotation)

Select the Plane. Follow these cook-book instructions.

With the Timeline Editor Cursor at Frame 1, turn **ON** Auto Keyframing.

Press **I Key** . Select **Rotation** in the Keyframe menu. A Keyframe is manually entered at frame 1 in the Timeline Editor.

With the Timeline Cursor at frame 30 and the Mouse Cursor in the 3D Viewport Editor, press R Key + Z Key + 90 + Enter. Repeat this for Frames 60, 90 and 120 (Automatic Keyframes Insertion). When completed TURN OFF Auto Keyframing.

With the mouse Cursor In the Graph Editor press **Shift + E Key** and select **Linear Extrapolation** to smooth the rotation.

Playing the animation will show the Plane continually rotating about the Z Axis.

Add a Particle System

Add a Particle System to the Plane with Particles being Emit from **Vertices** (Source Tab) and **Rendering as** the **Ico Sphere** Object (Render Tab – Render as Object – Object Tab – Instance Object - Icosphere). Play with the Scale and Randomness settings. Increase the **Lifetime to 250** and in the **Source Tab** uncheck **Random Order. Turn off Gravity**.

Playing the simulation (animation) in the Timeline Editor show Particles displayed as Ico Spheres being emitted and rising in spirals as the Plane rotates (Figure 24.93).

This is a relatively simple exercise demonstrating how an effect is created by combining the different Particle settings. There are no hard and fast rules which have to be obeyed. The way to create something is to experiment and

when you produce a worthwhile result, save the file and record results for future use. The following example incorporates more combinations.

24.19 More Arrays

Particle Arrays are only limited by your imagination and your knowledge of what tools to use and where the tools are located. This example will help you on your way.

In the default Blender Scene replace the default Cube with a **UV Sphere** as a **Particle Emitter** and add an **Ico Sphere** to be used as a Rendered Object (Instance Object).

Give the Ico Sphere Smooth Shading and park it to one side of the Scene. Scale down to approximately 0.250. The Ico Sphere will be what is termed the **Instance Object**. When Particles are generated by the UV Sphere they will display as this Object.

Add Lamps to improve illumination in the Scene.

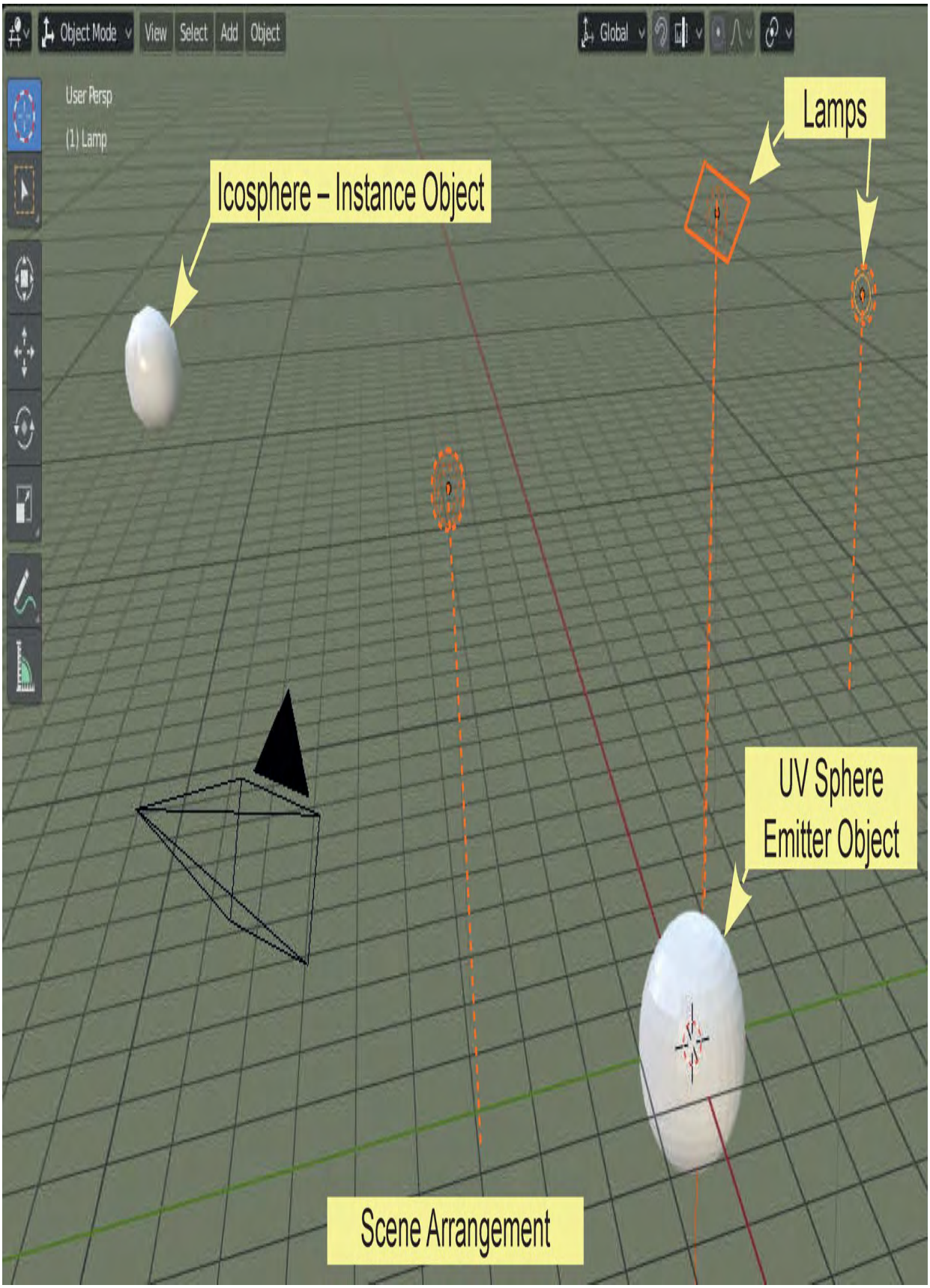

Object Mode
View
Select
Add
Object
Global
User Persp
(1) Lamp
Lamps
Icosphere – Instance Object
UV Sphere
Emitter Object
Scene Arrangement

Figure 24.94

The objective in the exercise is to create an array of small Objects that could represent a swarm of insects hovering around the UV Sphere or a cloud of stars in a far off galaxy. This is where your imagination comes into play. For realism you would create a small model of what the Objects in the Array were to be but for simplicity, the Ico Sphere will be used.

Turn off **Gravity** in Scene buttons to make the Particles disperse in 3D Space. Select the UV Sphere and add a Particle System.

Playing an animation in the Timeline Editor at this point with the default Particle System would see particles being emitted as small circles which float away from the UV Sphere and disappear after 50 Frames from their point of creation (Lifetime 50.000). To generate something a little more exciting modify the Particle System.

In the Emission Tab

Increase **Number** to 30000.
Change Lifetime to 999 (i.e. Forever).
Leave Start: 1 and End: 200.000.
Change Lifetime Randomness to 0.800

In the Render Tab

Change Render As from Halo to Object.
Leave Scale at 0.050
Change Scale Randomness to 0.080.

In the Render, Object Tab

Click where you see **Instance Object** and select **Ico Sphere**.

In the Timeline Editor play the animation then move the Timeline Cursor to Frame 120 and rotate the Viewport.

Figure 24.95 shows the Array generated in Rendered Viewport Shading Mode. The Array is ordered as seen by the Particles radiating out from the UV

Sphere as scattered lines. By zooming in on the 3D Viewport Editor you will see that each Particle is an Instance (copy) of the Ico Sphere parked in the Scene.

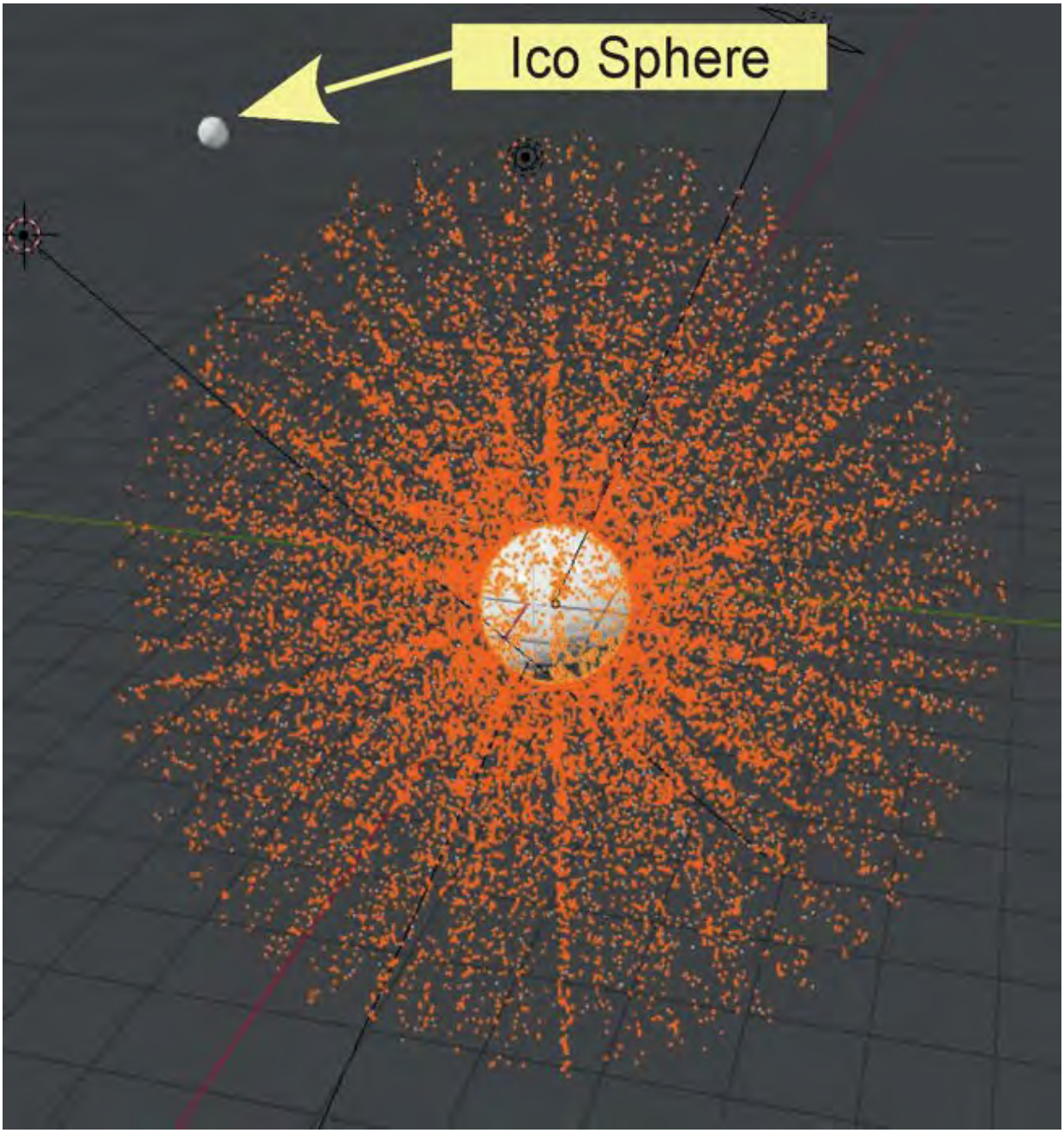


Figure 24.95

From this point you make further modifications to settings to alter the appearance of the Array.

Add a Material to the Ico Sphere.

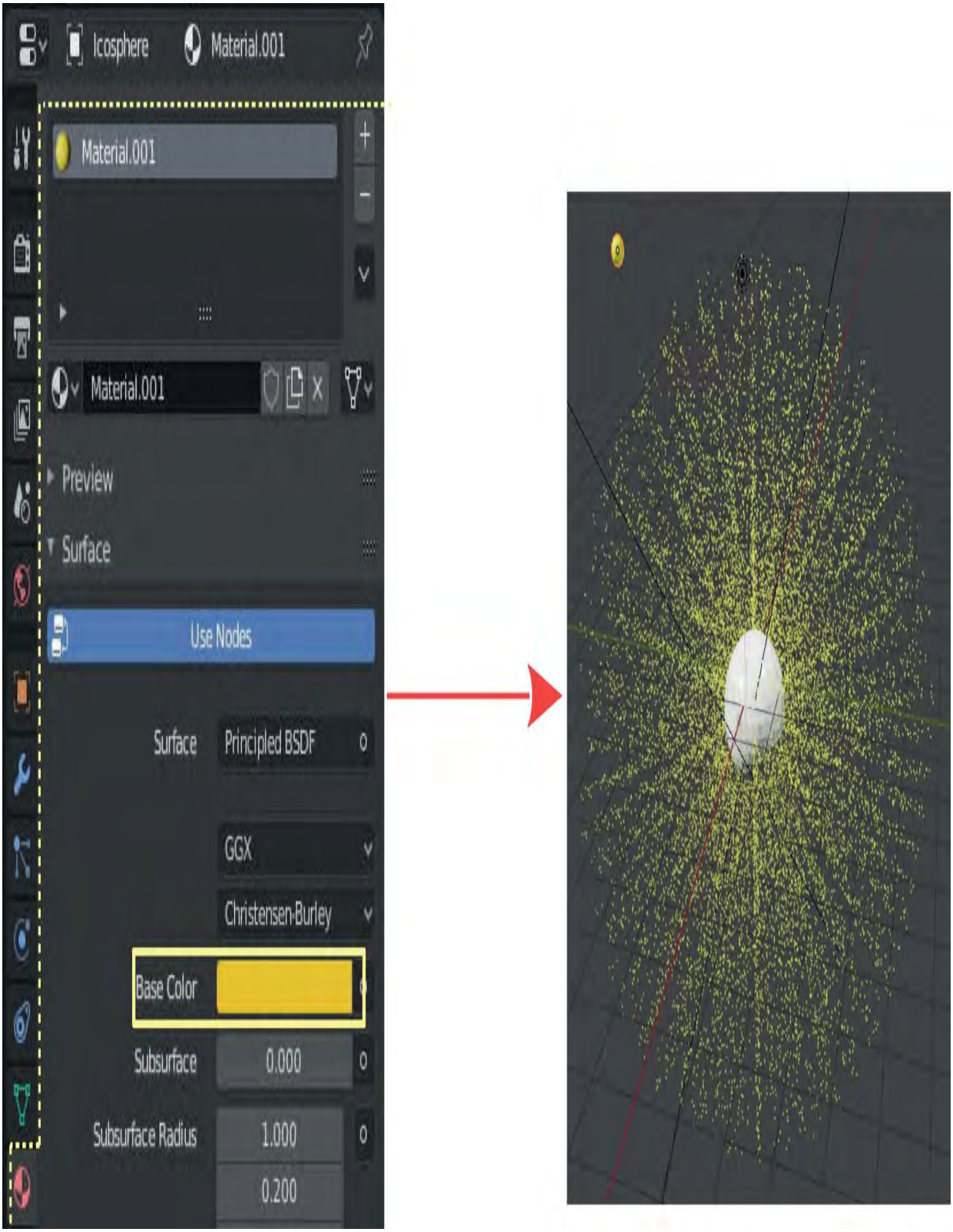

Icosphere
Material.001
Material.001
Material.001
Preview
Surface
Use Nodes
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
1.000
0.200

Figure 24.96

To see the Material be in **Material Preview Viewport Display Mode**.

Deselect the UV Sphere and add an **Empty Object** and activate a **Force Field, Type Turbulence,** Strength 5 in **Physics Properties** (Figure 24.98).

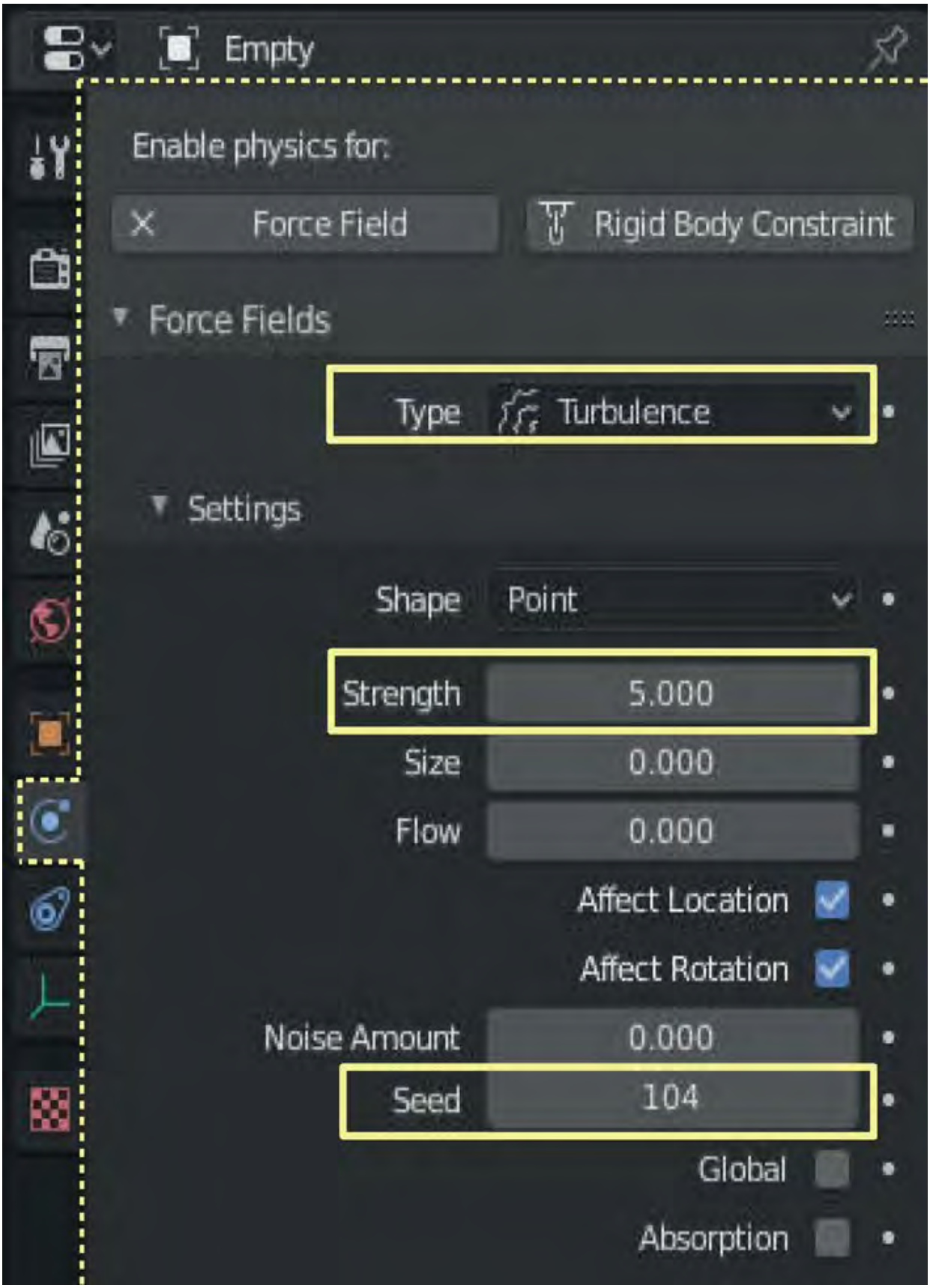

Empty
Enable physics for:
Force Field
Rigid Body Constraint
Force Fields
Type
Turbulence
Settings
Shape
Point
Strength
5.000
Size
0.000
Flow
0.000
Affect Location
Affect Rotation
Noise Amount
0.000
Seed
104
Global
Absorption

Figure 24.98

Play animation to see flow of Particles.

Particles Dispersed by Force Field Figure 24.99 shows a Screen Capture taken while playing the animation at approximately Frame 120.

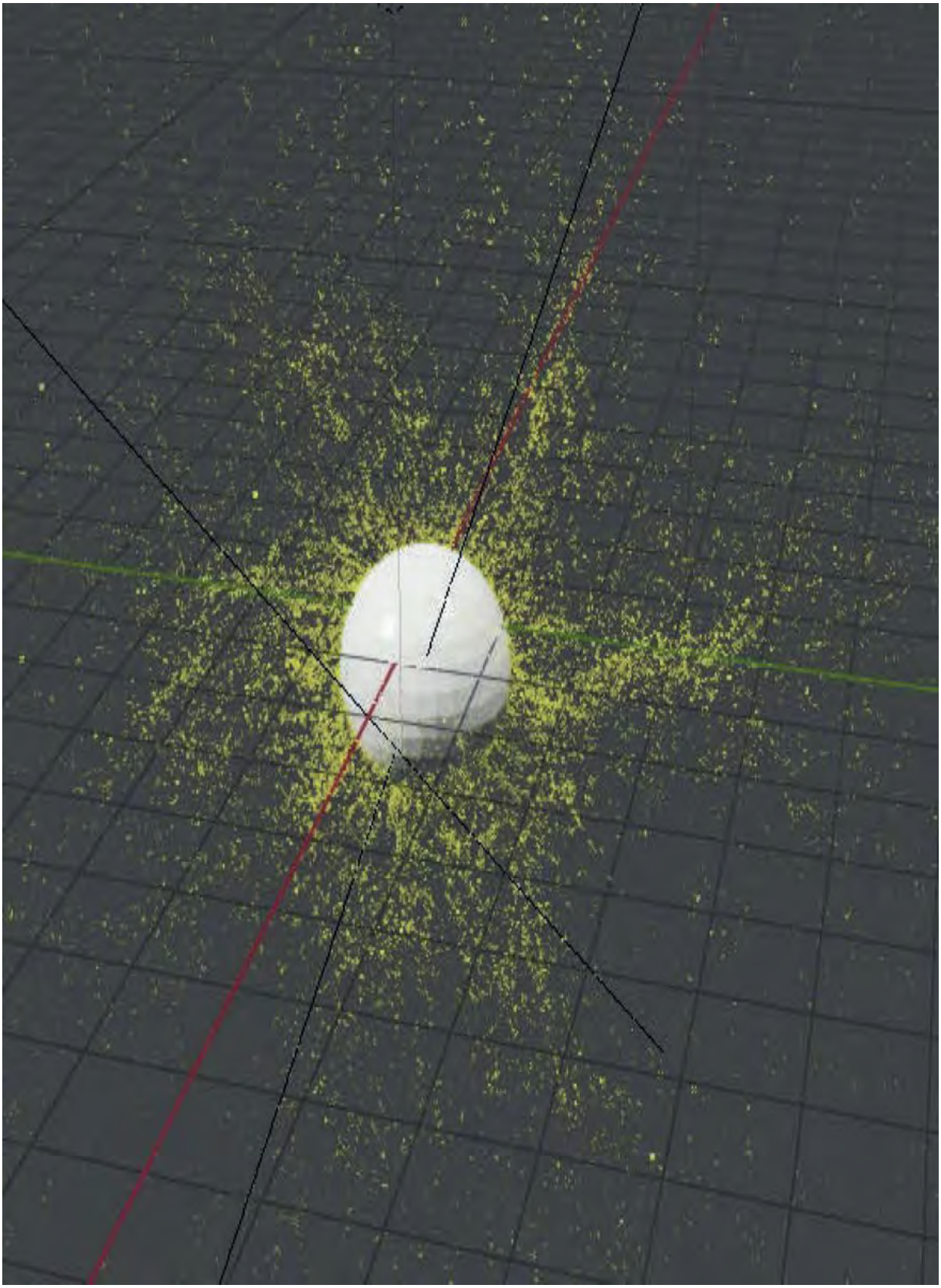


Figure 24.99

Select the **UV Sphere** and change the **Particle System to Emitter Type: Hair**. With the animation in the Timeline Editor at Frame 1 the UV Sphere displays as a speckled orange / yellow disk.

In the **Particle System, Render Tab,** change **Render As** to **Path**. In the Children Tab select Simple for a different sort of Array (Figure 24.100).

Figure 24.100

The possibilities are endless, therefore, experiment and record settings or save Blender files for future use.

24.20 The Assignment Panel

When a **Particle System** is first added to a Scene by clicking on the plus sign in the Particles Properties, Blender introduces data to the Scene that creates a default Particle System. Blender names this data block **ParticleSettings,** as seen in the **Settings panel**. The data block named **ParticleSettings** is automatically linked to the default Particle System that is named **ParticleSystem**. ParticleSystem is placed in the Assignment Panel where it is assigned to an Object in the 3D Viewport Editor.

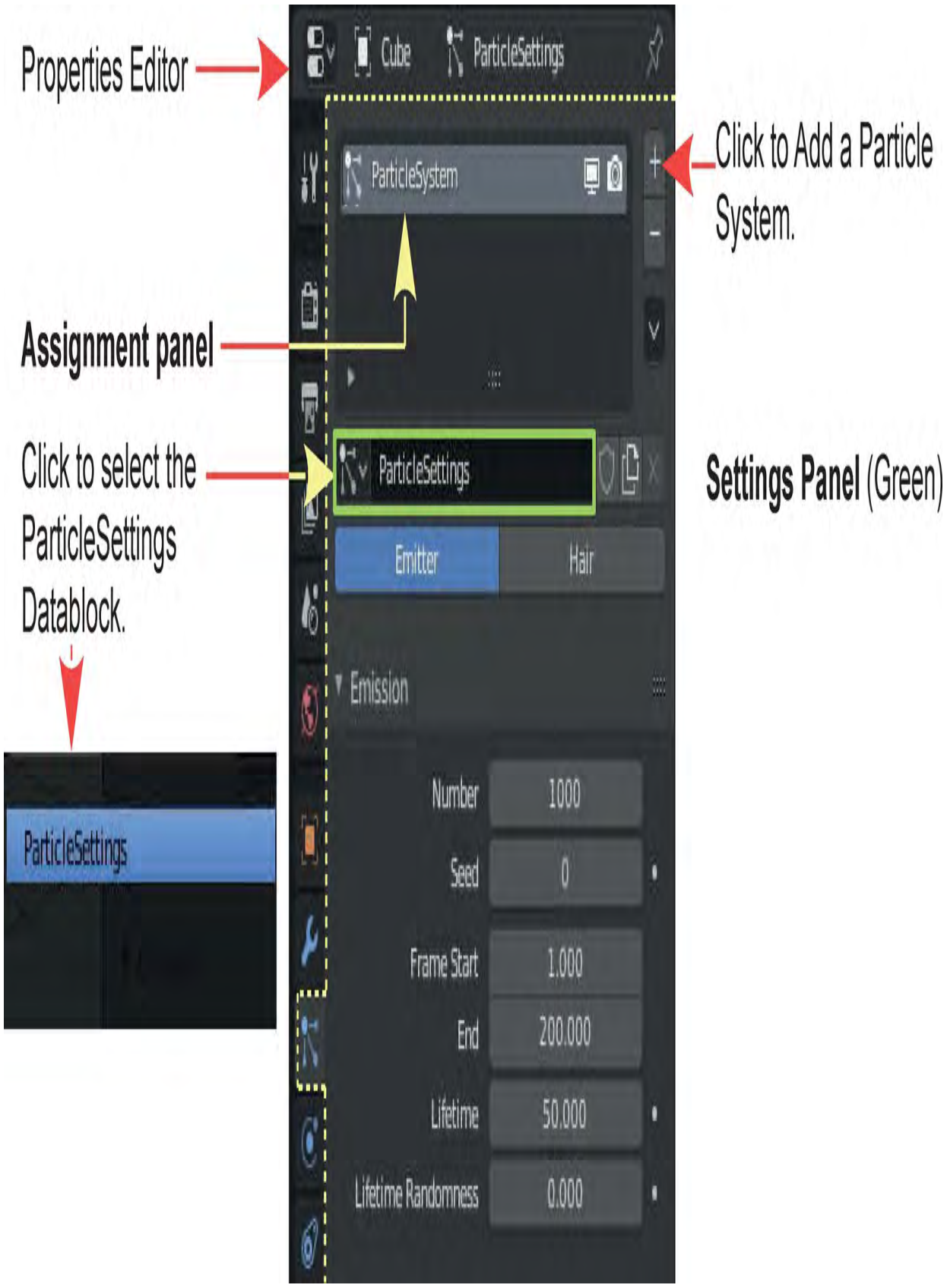

Properties Editor
Cube
ParticleSettings
ParticleSystem
Click to Add a Particle System.
Assignment panel
Click to select the ParticleSettings Datablock.
ParticleSettings
Settings Panel (Green)
Emitter
Hair
Emission
Number
1000
Seed
0
Frame Start
1.000
End
200.000
Lifetime
50.000
Lifetime Randomness
0.000
ParticleSettings

Figure 24.101

There is no **Assignment Tab** or **Assignment Panel** as such, but for the purpose of this discussion consider the **Settings Panel** marked in green as the **Assignment Tab** and the panel displaying **ParticleSystem** as the Assignment Panel (Figure 24.100).

Below the Settings panel there is a Particle Type selection menu with the two options, **Emitter** and **Hair**.

Type: Emitter is the default selection, which means that with the Particle System assigned to an Object in the Scene, that Object becomes the emitter of the Particles. In either case, the Object becomes an emitter with a Particle System assigned. **Type: Hair** may be viewed as a specialised static emitter which generates Particles that Render.

In Figure24.100, Particle System, Type: Emitter is selected with the ParticleSettings Datablock. assigned to the ParticleSystem which in turn is assigned to the Object which is selected in the 3D Viewport Editor.

Note that the names **ParticleSettings** and **ParticleSystem** may be renamed by double clicking in the panels, deleting the name and retyping a new name. This is useful when there are multiple Objects, Data Blocks and Particle Systems. Multiple Objects in the 3D Viewport Editor can each have a different Particle System assigned, and each Object may have more than one Particle System.

When a new Particle System Data Block is added to the Scene, Blender creates a new name for the Data Block. The default particle settings Data Block is named **ParticleSettings** as previously stated. When a second Data Block is added, it is named **ParticleSettings.001,** a third would be named **ParticleSettings.002,** etc.

Renaming Data Blocks to something more relevant to Objects in the Scene would be an advantage. When new Data Blocks are created, they are stored in

a cache for reuse by other Particle Systems.

When a new Particle System is added to the Scene, Blender assigns that system to the Object selected in the 3D Viewport Editor. If no Objects are selected, the new Particle System is assigned to the last Object that was introduced to the Scene. Particle Systems added to a Scene initially have the default ParticleSettings Data Block linked and a new name applied as described previously. At this point, the Data Block settings may be altered to create a new unique Data Block or a previously created Data Block may be selected and linked to the new Particle System. Clicking on the icon in front of the Particle Settings panel reveals a drop down menu showing the Cache, mentioned previously, with Data Blocks for selection.

The forgoing statements may seem confusing and not easily related to what has been labelled the **Assignment Tab**. The following exercise will attempt to clarify the statements and at the same time demonstrate the application of Particle Systems in practical terms.

24.21 Paticie Exercises

Open a new Scene in Blender and delete the Cube from the 3D Viewport Editor. Add three separate Plane Objects and position them at the center of the Scene so that they are all visible in Camera View . Add a diffuse Material color to each of the Planes. Make the colors red, green and blue. (Figure 24.101). To see Materials in the 3D Viewport Editor be in Material Preview Viewport Shading Mode.

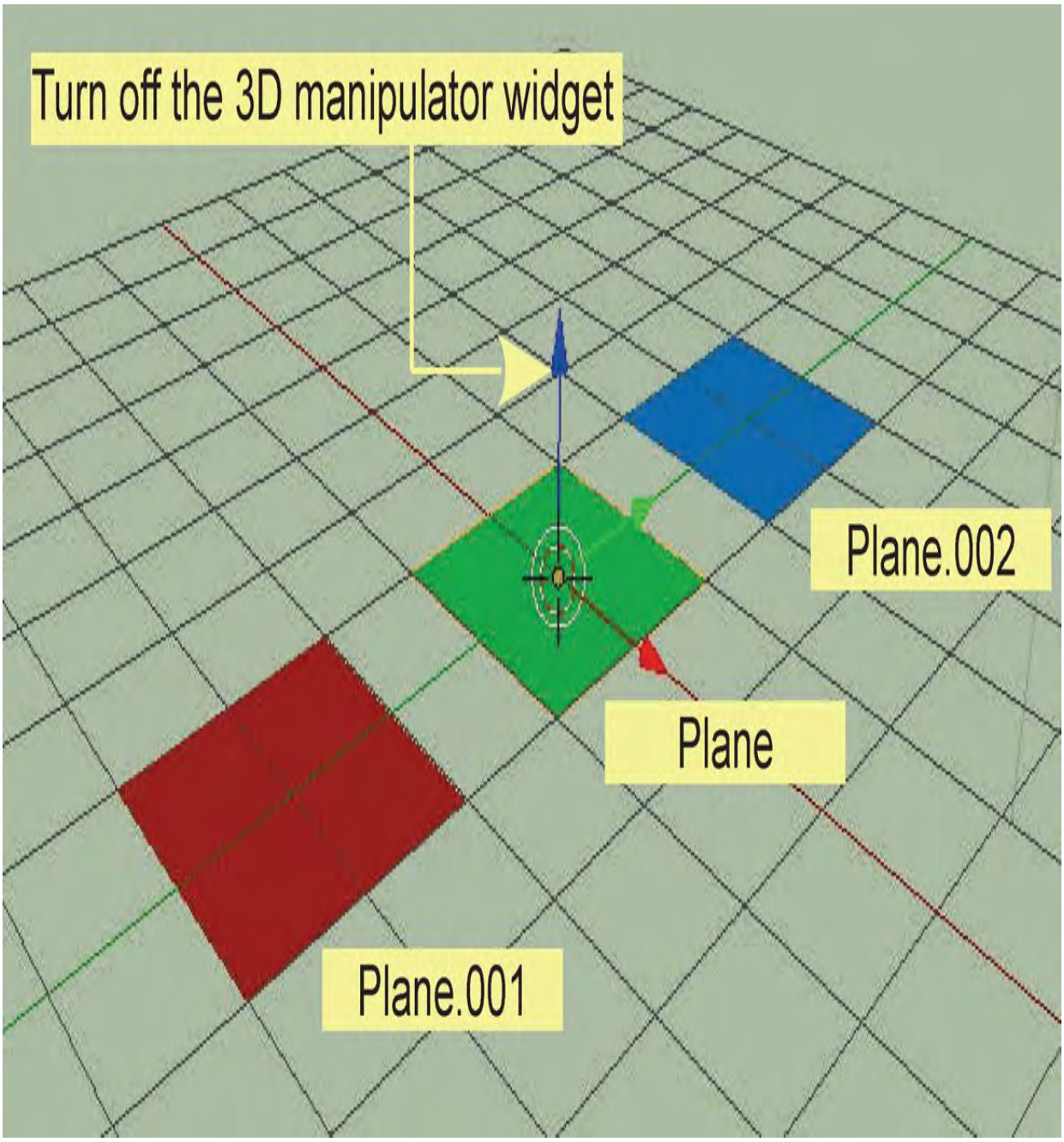


Figure 24.102

Turn off the **Gravity** setting in the Properties Editor, Scene Properties, Gravity tab (untick).

Turn off the 3D Manipulator Widget in the 3D Viewport Editor.

At this time, the three Plane Objects have been named **Plane, Plane.001,** and **Plane.002** by Blender, as seen in the upper left-hand corner of the 3D

View Editor when each is selected. In the diagram the green Plane was entered first followed by the red Plane then the blue Plane.

This automatic naming is not all that relevant to what is in the Scene, therefore, the Planes will be renamed.

In the 3D Viewport Editor, select the red Plane and go to the **Properties Editor, Object Data Properties**. At the top of the Editor you will see **Plane.001** in the Data Block ID name panel (Figure 24.103).

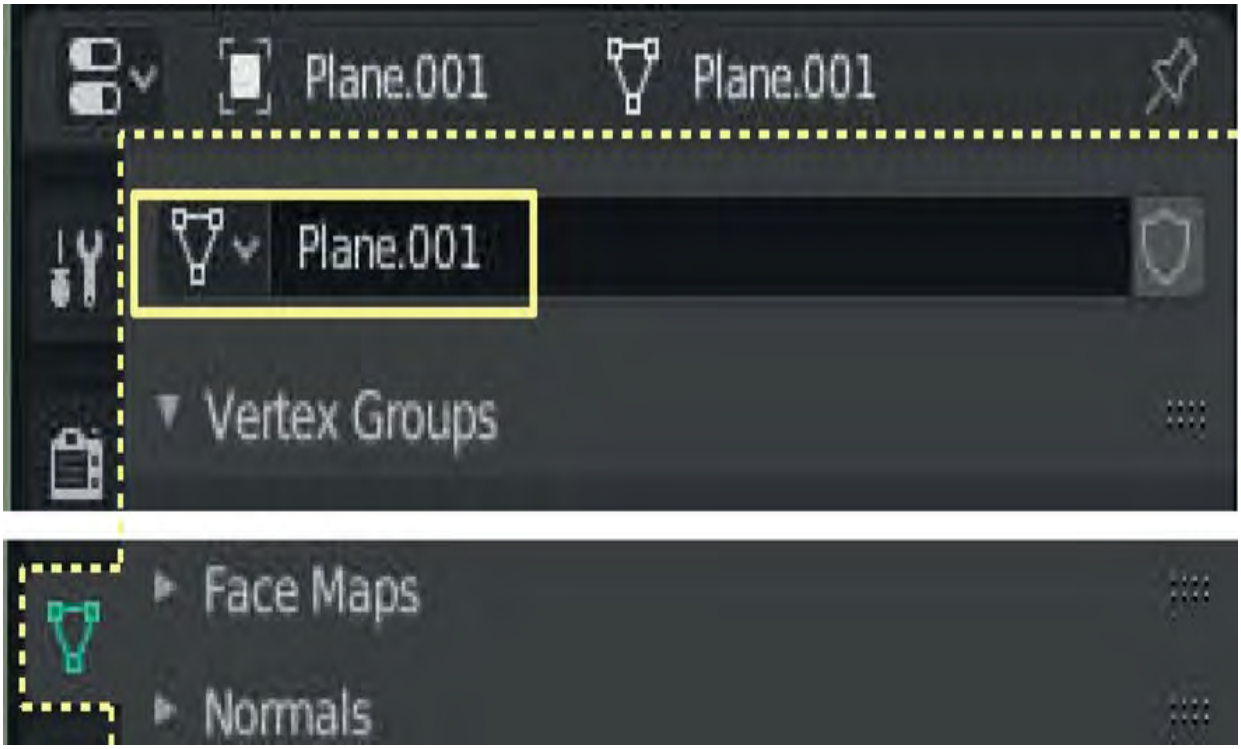


Figure 24.103

Click on the name (displays typing cursor), hit delete, type in **Red_Plane** and press Enter. Select the green plane in the 3D Viewport Editor and rename it **Green_Plane,** and then similarly for the blue Plane.

Note: Renaming may be done in the Properties Editor, Object Properties, Datablock ID name panel or in the Outliner Editor (upper RH

of the Screen).

Add **Particle Systems** to the Planes. Select the red Plane and click on the **Particles Properties** in the **Properties Editor**. Click **the plus sign** to add a Particle System. The Particle System panel displays with all the tabs and buttons for controlling the settings and has been set up with default values. Leave all the values as they are displayed except for the **Lifetime** and **Number** value in the **Emission Tab**. Change the value as shown in Figure 24.104. This will give you a better view of Particles being generated.

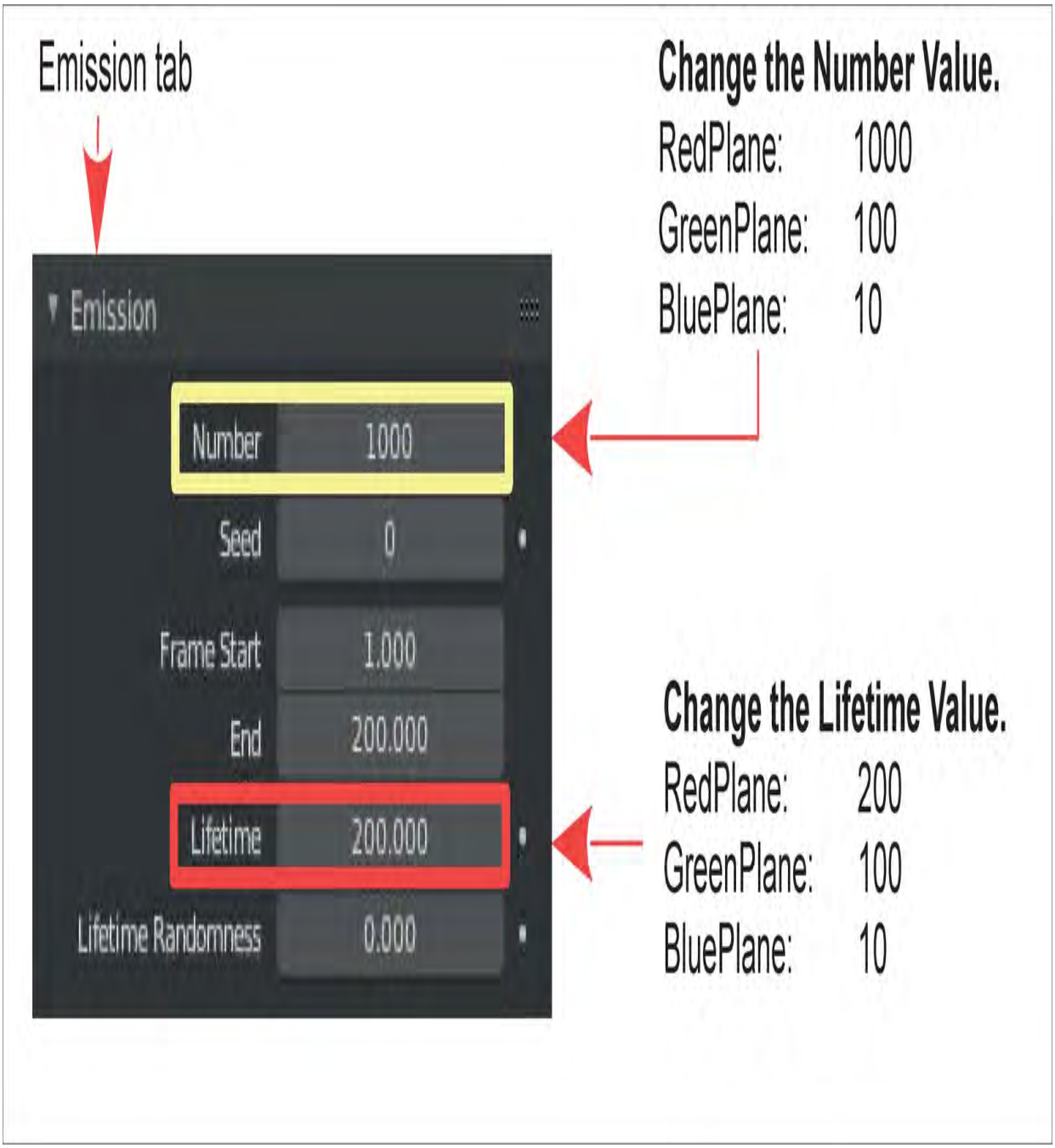
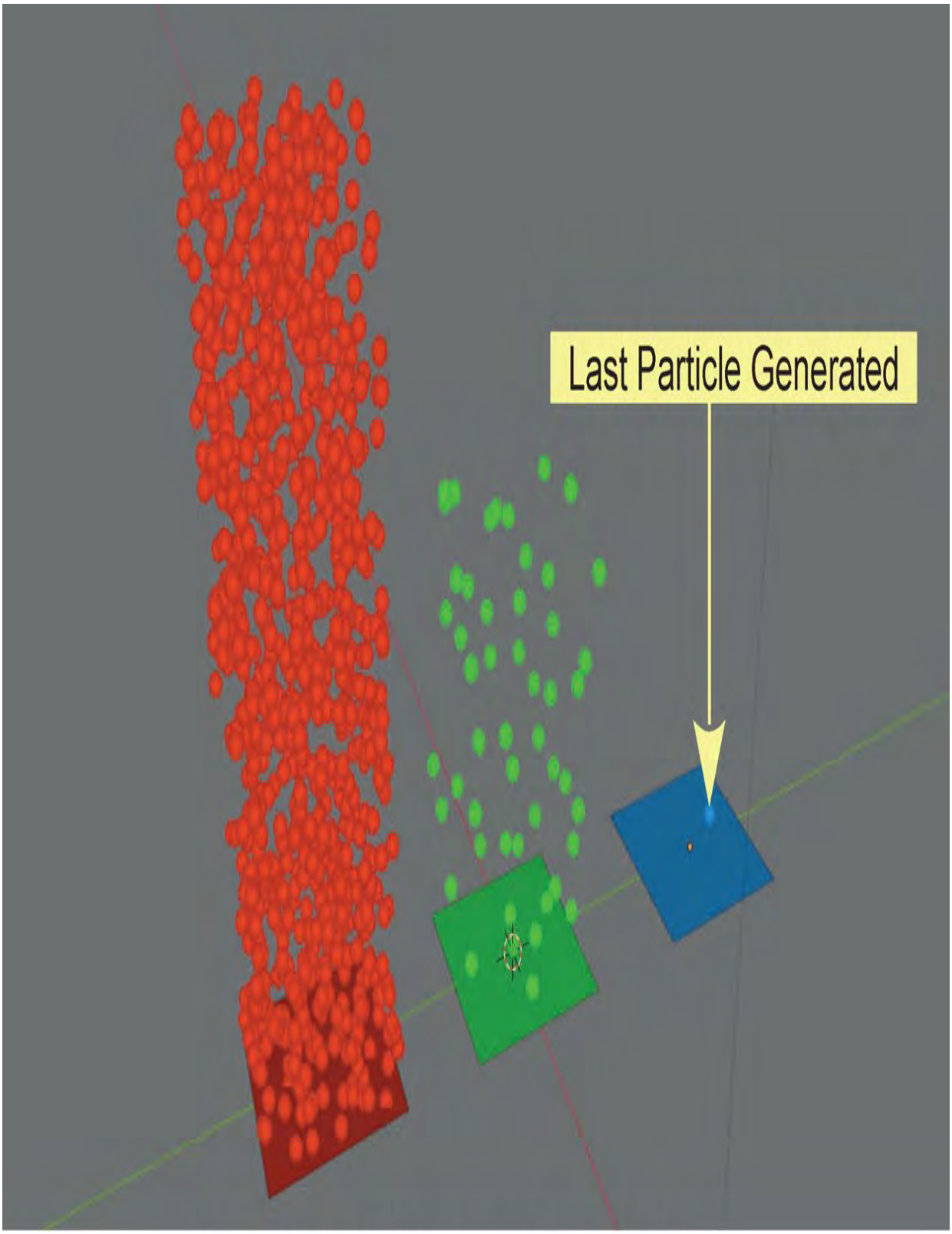


Figure 24.104

Do the same for the other two Planes. Play the animation in the Timeline Editor to show Particles being generated.

Cycle through the animation in the Timeline Editor (drag the Timeline Cursor) to frame 170 and observe the Particles (Figure 24.105). You have

three different Planes with three different Particle Systems–RedPlane: 1000 Particles, GreenPlane: 100 Particles, BluePlane: 10 Particles.



3D Viewport Editor at Frame 170

Figure 24.105

Blender generates Particles beginning at Frame 1. The number of Particles set for each Plane is generated and spread over 200 Frames (200 is the End value in the Emission tab). The actual length of the Animation is 250 Frames. In Figure24.105 BluePlane is showing the last of the 10 Particles being generated. For BluePlane the Particle Lifetime is set at 10 Frames, therefore, the first Particle disappears at Frame 11, the next appears at Frame 21 and disappears at Frame 31. For the BluePlane the last Particle appears at Frame 161 and disappears at Frame 171.

In the 3D Viewport Editor select each Plane separately and note the names that display in the panels in the **Properties Editor, Particles Properties , Assignment tab,** (Figure 24.106).

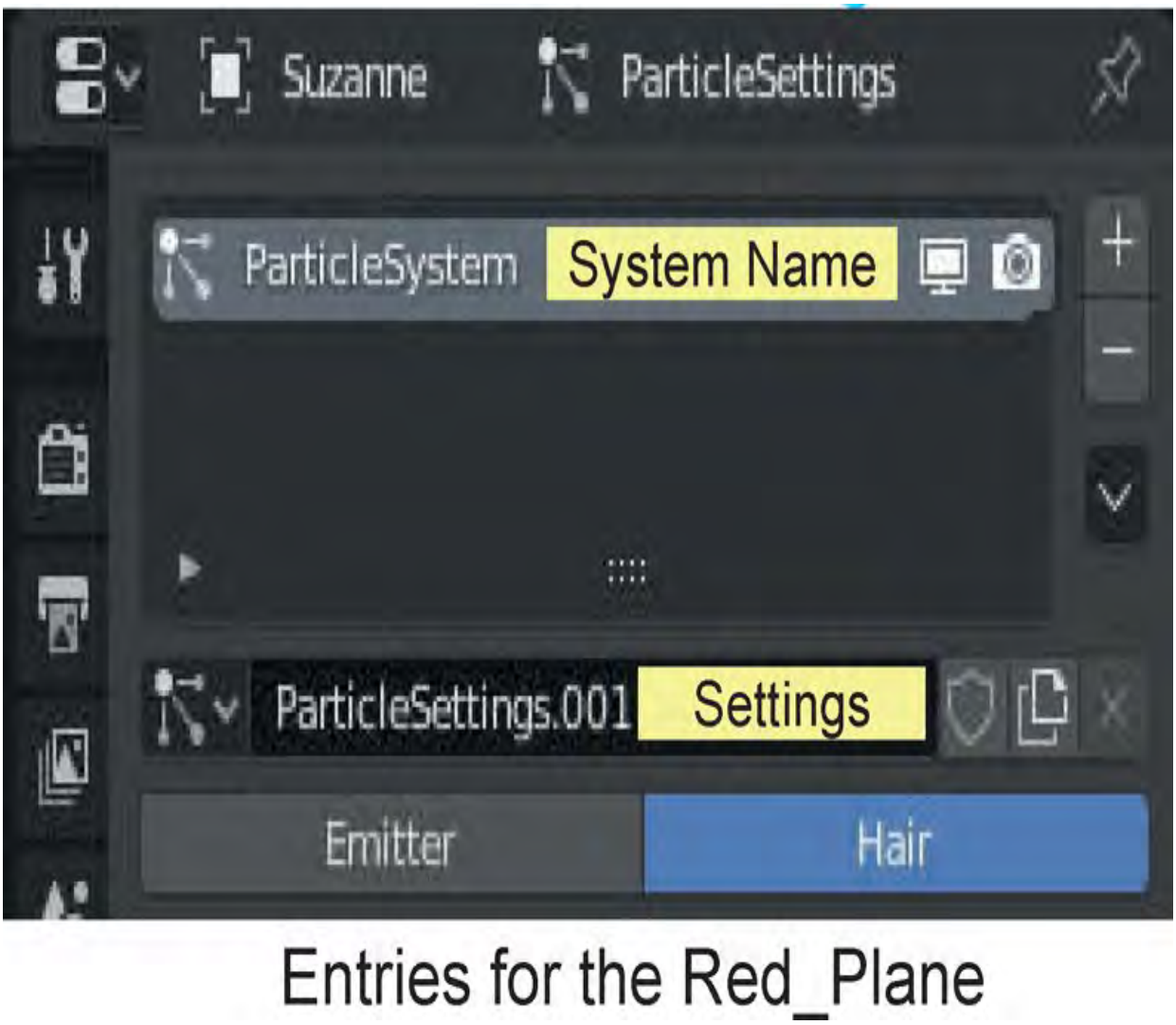


Figure 24.106

It was previously stated that there were three separate Particle Systems, however, you see that the three names are all **ParticleSystem,** but each one has a different **Settings** name. At this stage it's probably a good idea to do some renaming.

- **Red_Plane**
  - Name: ParticleSystem
  - Settings: ParticleSettings
- **Green_Plane**

– Name: ParticleSystem
– Settings: ParticleSettings.001

- **Blue_Plane**
  – Name: ParticleSystem
  – Settings: ParticleSettings.002

Change the names to the following:

- **Red Plane**
  – Name: RedPSystem
  – Settings: RedPSettings
- **Green_Plane**
  – Name: GreenPSystem
  – Settings: GreenPSettings
- **Blue_Plane**
  – Name: BluePSystem
  – Settings: BluePSettings

After renaming, proceedings should be easier to follow.

To continue; In the 3DView Editor select the **Green Plane** to reassign some settings.

In the **Properties Editor, Particles Properties, Assignment Tab**, click on the button just in front of the name panel and next to Settings.

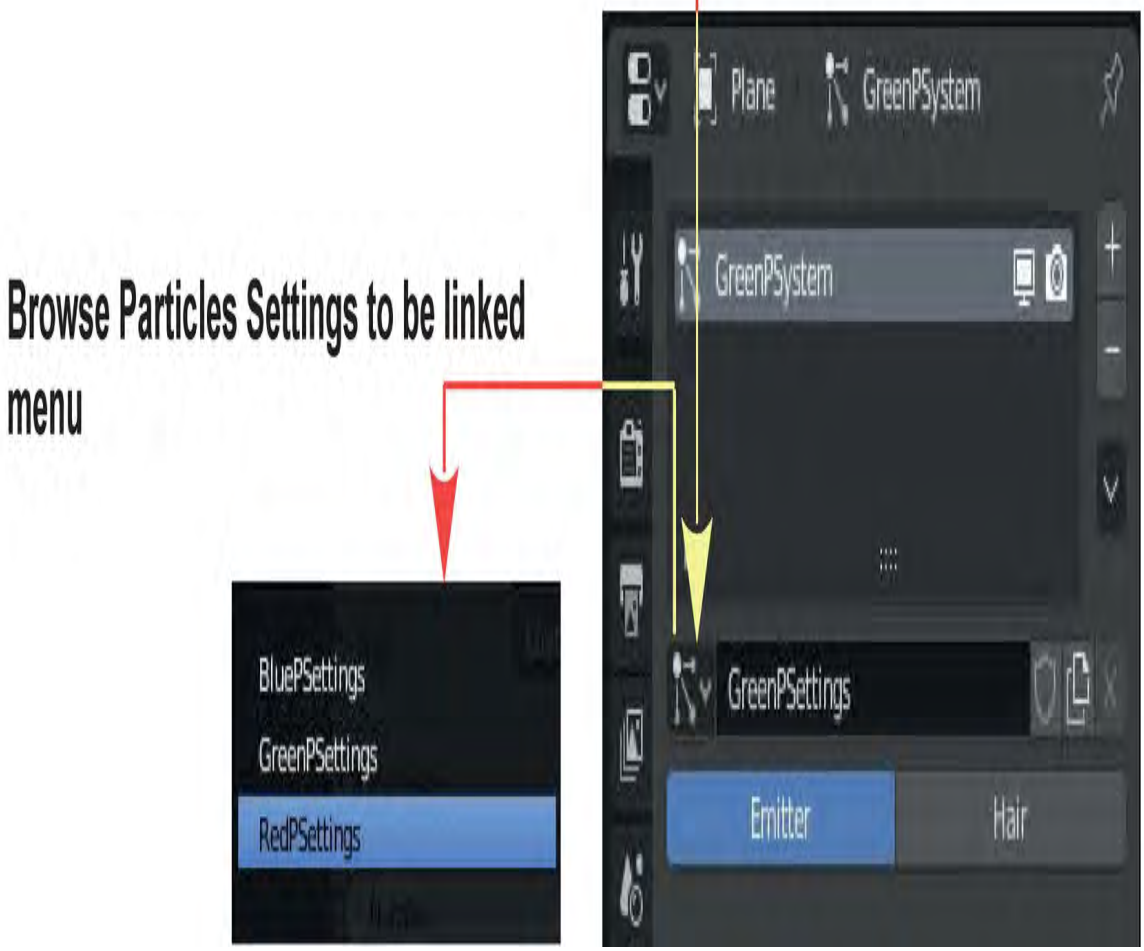


Figure 24.107

The menu that displays has the names of the three **Particle Settings Data Blocks** (Figure 24.108). Whenever a new group of Particle Settings is created, Blender puts it into a cache for reuse. You can see these **Data Blocks** in the **Outliner Editor** in **Data API Mode**.

Figure 24.108

With the **BluePlane** selected in the 3D Viewport Editor, click on **GreenPSettings** in the **Browse Particles to be linked menu**.

You will have the **BluePSettings** assigned to the **GreenPSystem**.

If you replay the Particle generation animation, the green and blue Planes generate the same number of Particles.

Note that in the **Settings panel** for the green and the blue Planes, a number 2 has appeared; this tells you that **BluePSettings** is being used by two systems. The number of Particles Emitted is set by the Particle System settings.

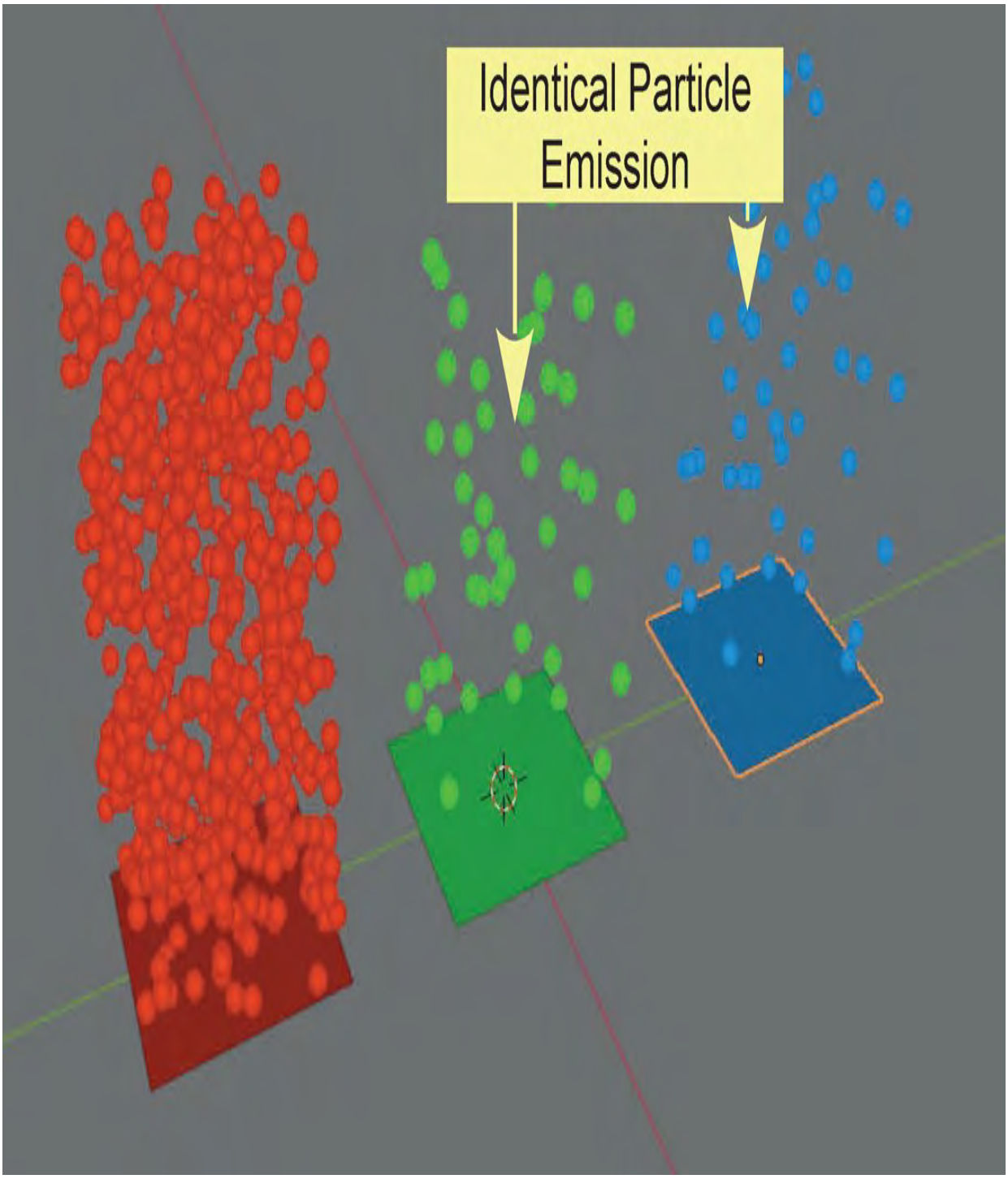


Figure 24.109

The forgoing has demonstrated that any Settings Data Block can be assigned to any Particle System.

Continue by clicking on the number 2, which makes the Data Block a single user. Blender does this by leaving the original as it is and creating a new Data Block, however, the new Data Block is identical to the original.

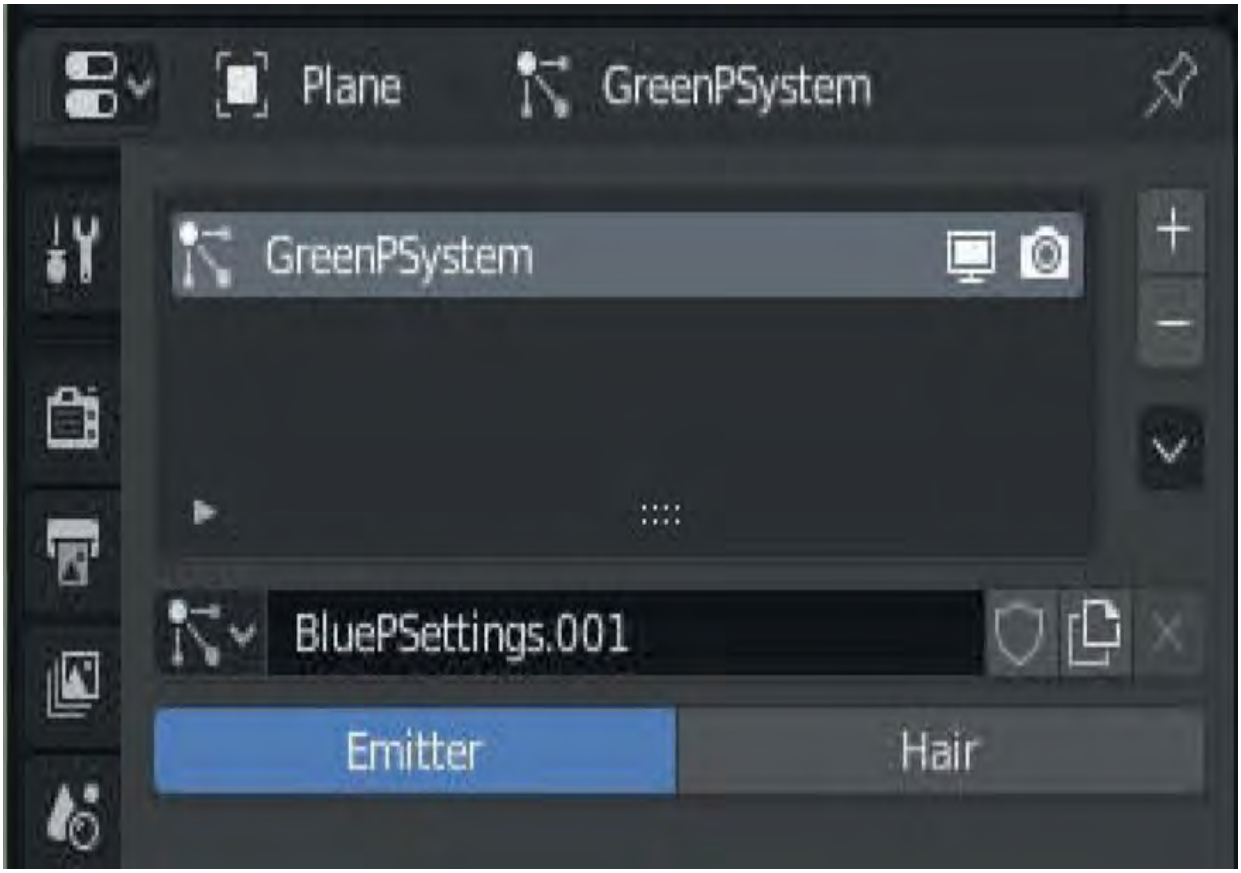


Figure 24.110

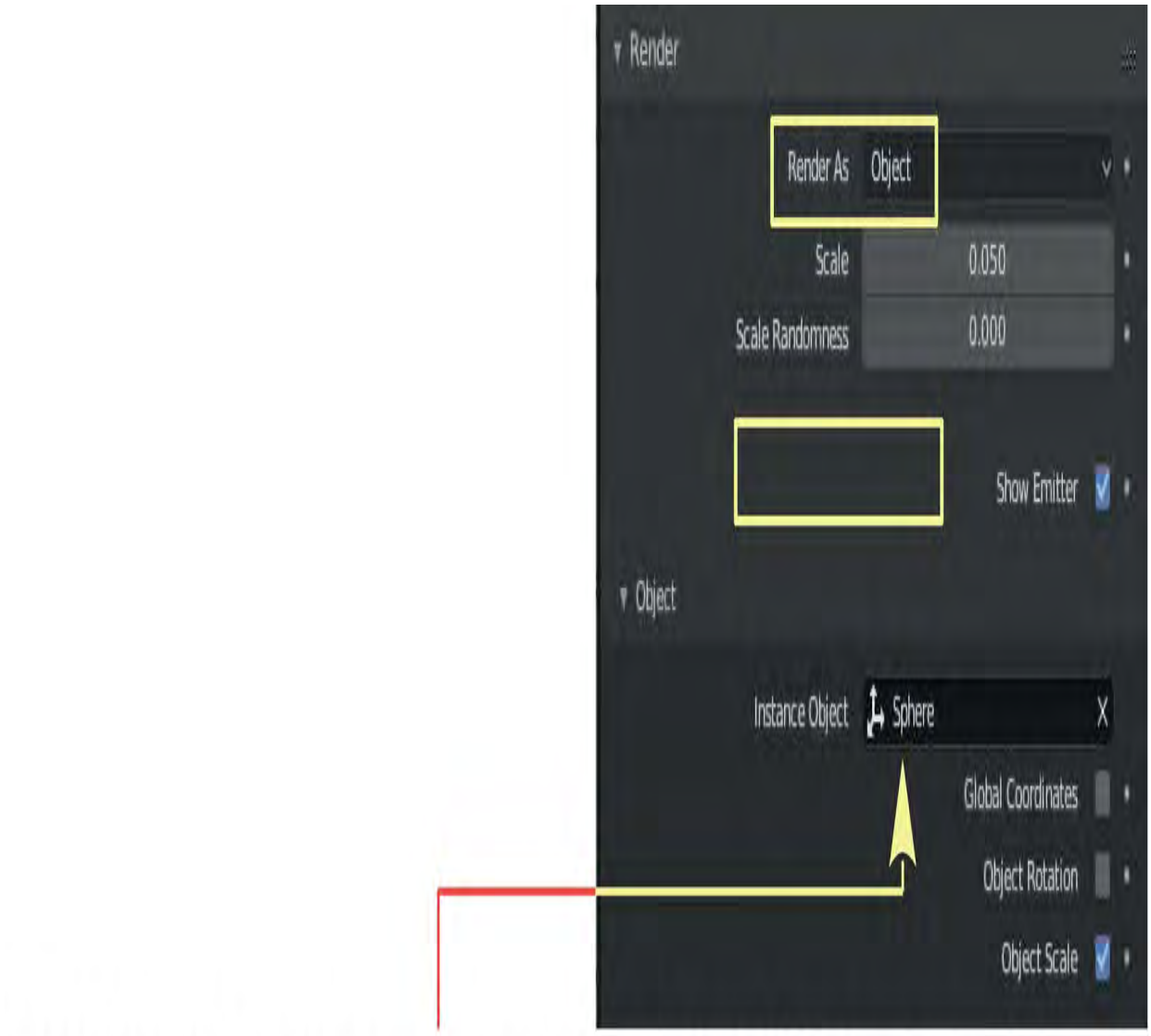
You can see that the settings name is **BluePSettings.001** (Figure 24.110).

In the **Emission tab** change the **Number** value to 10 and the **Lifetime** value to 30.

In the 3D Viewport Editor, add a **UV Sphere** to the Scene and give it a yellow Diffuse Material color.

Note: Blender has named the Sphere simply, **Sphere**. Make sure it is off to one side in the Scene away from the planes.

Select the Green Plane in the 3D Viewport Editor and then go back to the **Particle Properties** in the **Editor**.



In the **Object Tab** that displays, click where you see **Instance Object** and select **Sphere** (the yellow UV Sphere).

Figure 24.111

Change data in this new Data Block (BluePSettings.001) which is assigned to the GreenPSystem for the green Plane.

In the **Render Tab** in the **Particles Properties** (scroll down a bit). In the bar labelled **Render as** click and select Type: **Object** (Figure 24.111). Selecting **Type: Object** instructs Blender to Render (Display) Particles as an Object entered in the Scene.

You are telling Blender to display and Render the Particles as replicas of the yellow Sphere entered in the Scene. Play the particle generation animation and you will see yellow Spheres being generated (Figure 24.112). (increase the Render Scale Value).

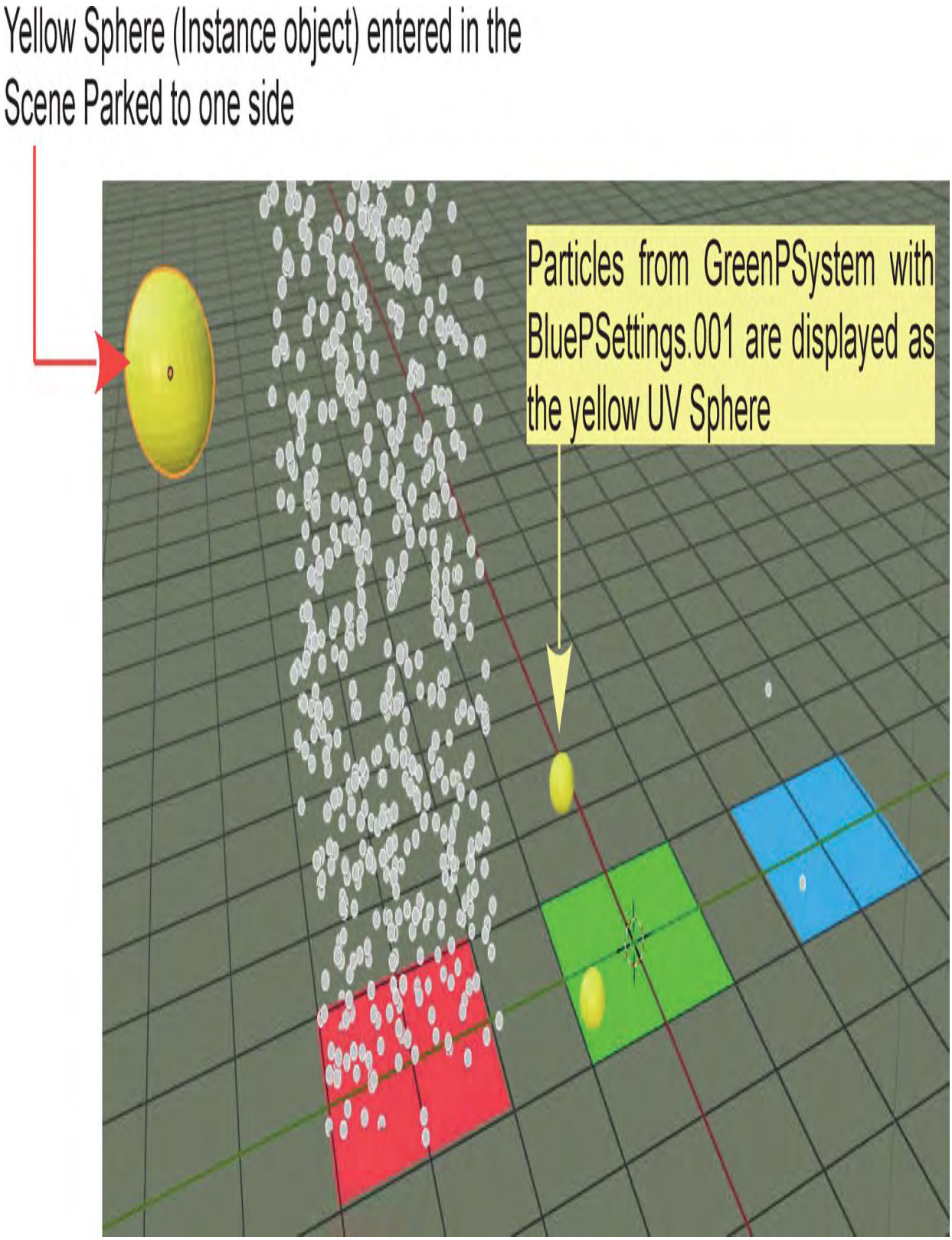


Figure 24.112

The size of the Spheres generated is determined by the size of the yellow Sphere entered in the Scene and the Scale value in the Particle Properties, Render Tab .

This exercise has demonstrated how manipulating values in the Properties Editor controls the Emission of Particles in the 3D Viewport Editor. This can be used to create visual effects in a Scene. **Note:** An Object can have more than one Particle System in operation at the same time.

24.22 Multiple Particle Systems

Work through the following exercise to see how to apply **Multiple Particle Systems**. A Plane Object is entered in the 3D Viewport Editor with a blue Material applied. The 3D Viewport Editor is in Material Preview Viewport Shading Mode.

A Particle System is added to the Plane in the Properties Editor, Particle Properties with the Emission values as shown in Figure 24.114 (default settings except Lifetime 200). Gravity has been turned off in the Scene buttons.

Playing the animation in the Timeline Editor produces the array of Particles shown in Figure 24.113.

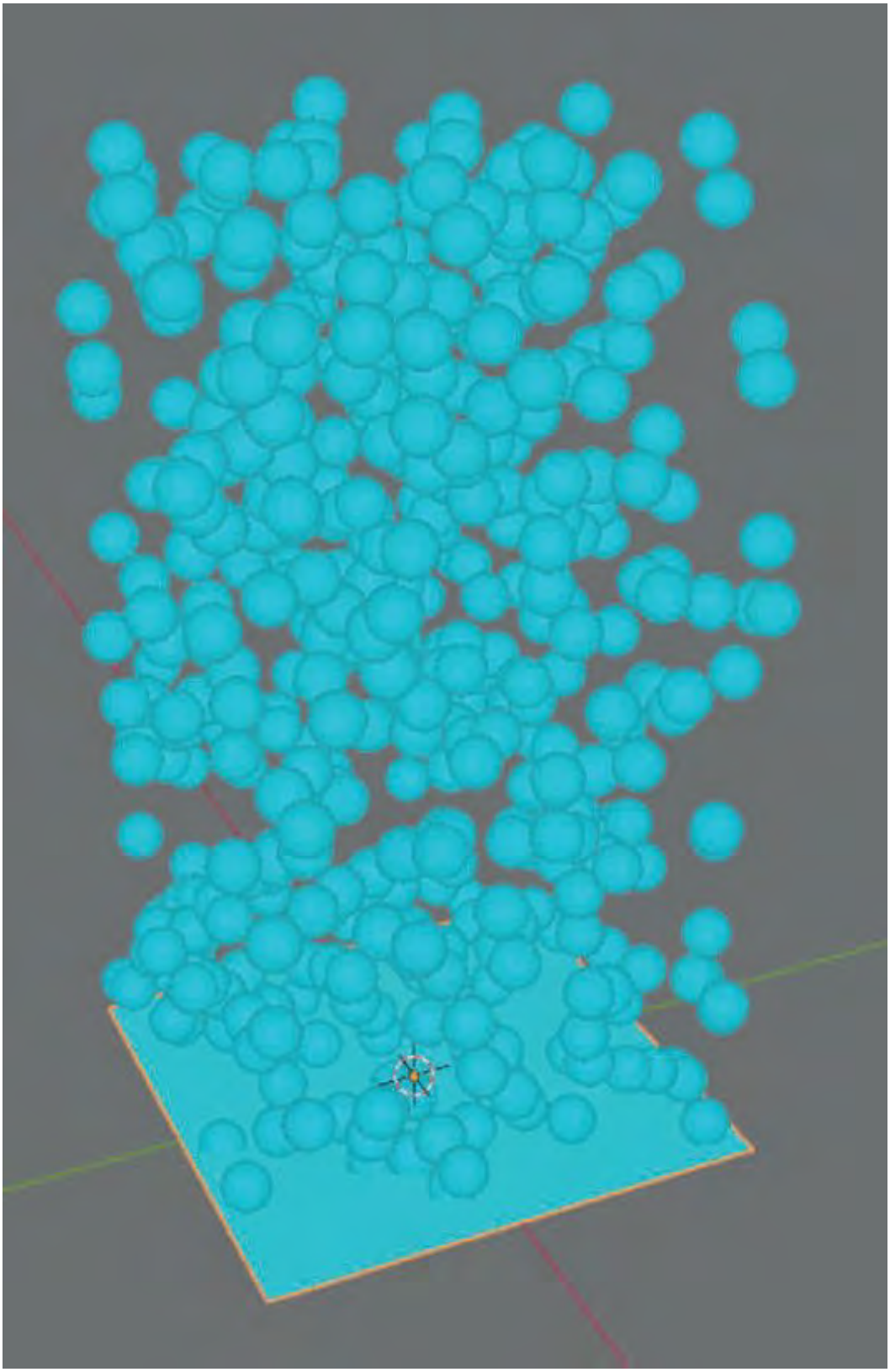


Figure 24.113

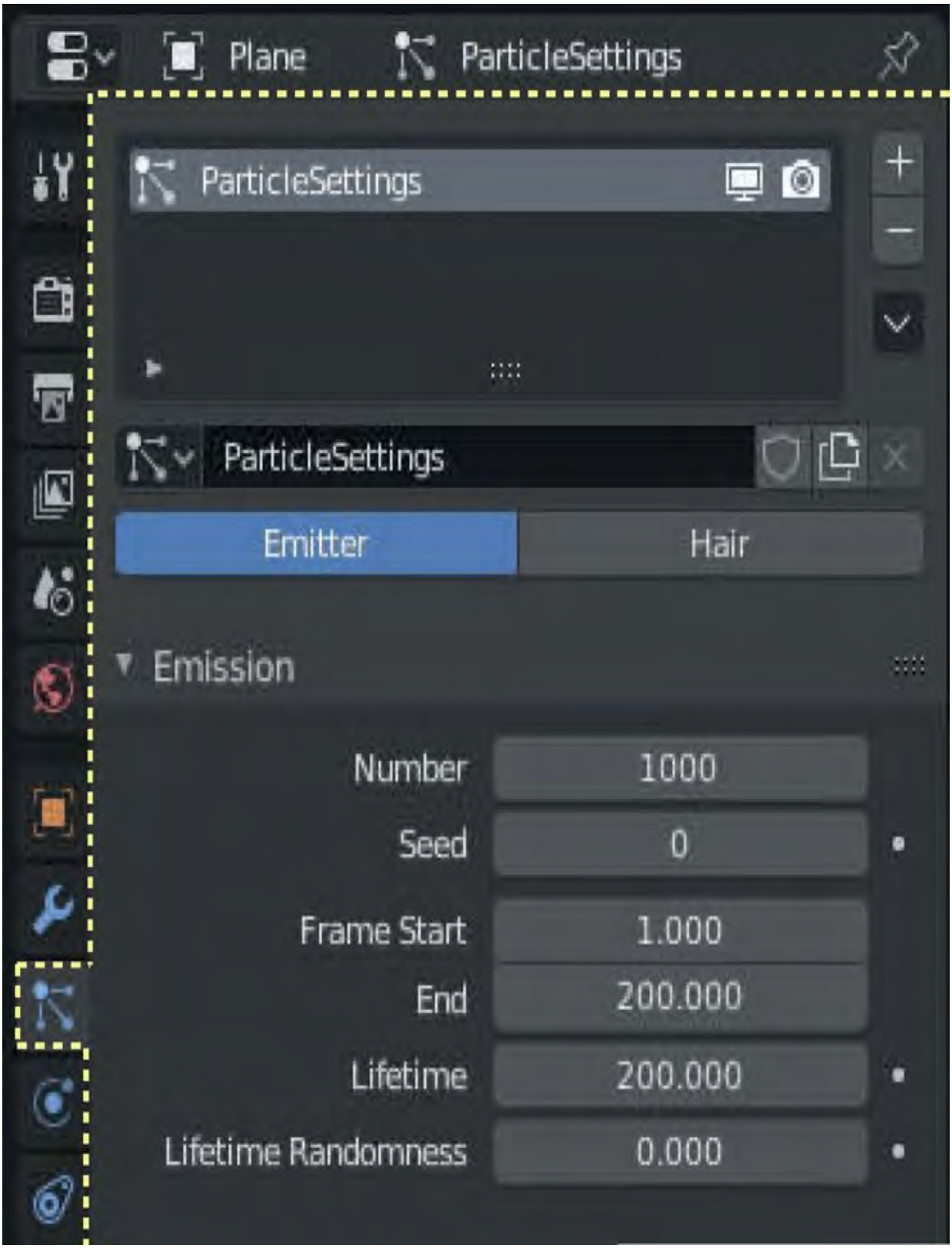


Figure 24.114

You may have the Particles display as another Object. Add a UV Sphere to the Scene. Give the Sphere a nice bright Material (color) and park it off to the side. **Select the Plane**.

In the **Particle System,Render Tab** change **Render As** to **Object** (Figure 24.115) then in the Object Tab select Sphere as the **Instance Object**.

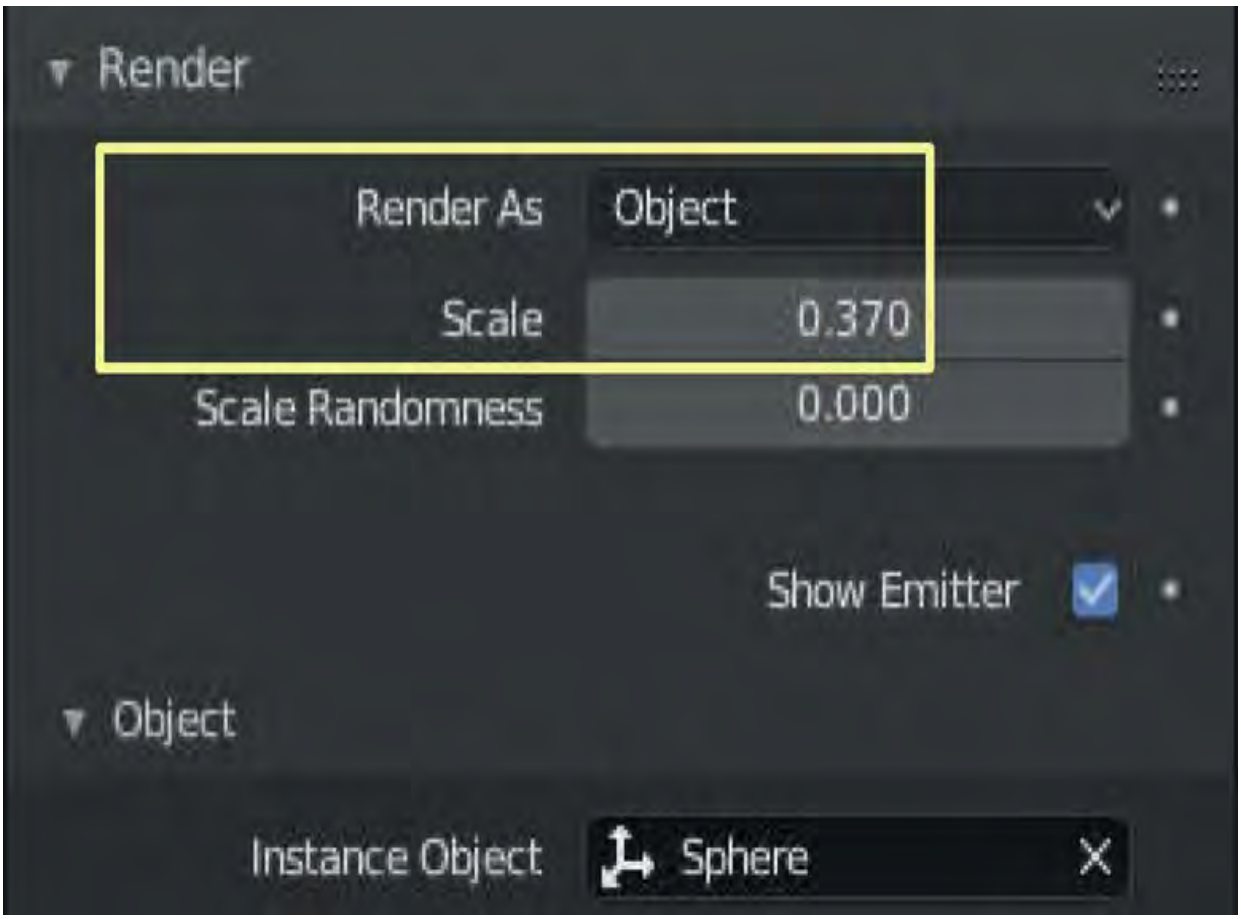


Figure 24.115

Replay the animation For a colorful display (Figure 24.116).

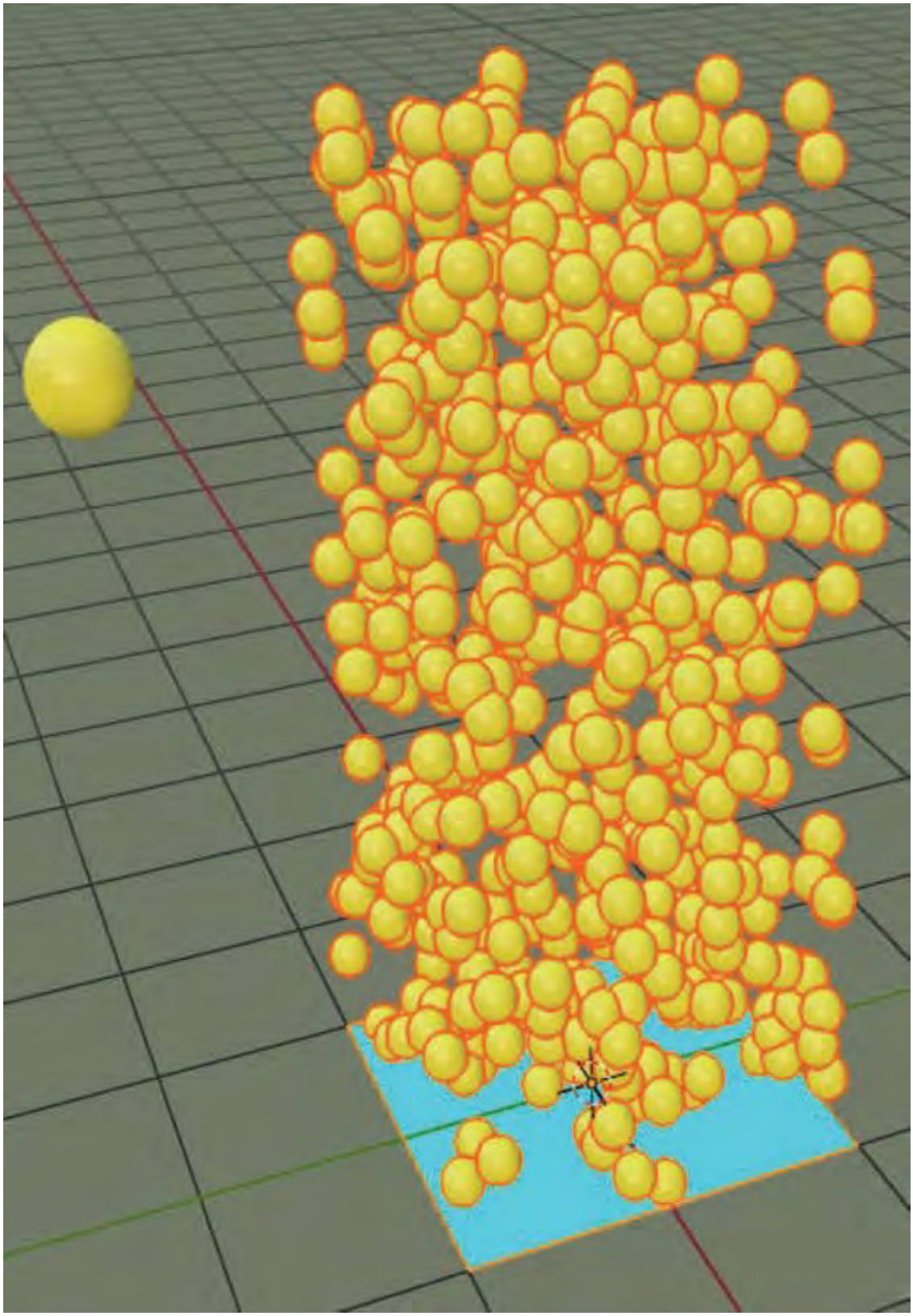


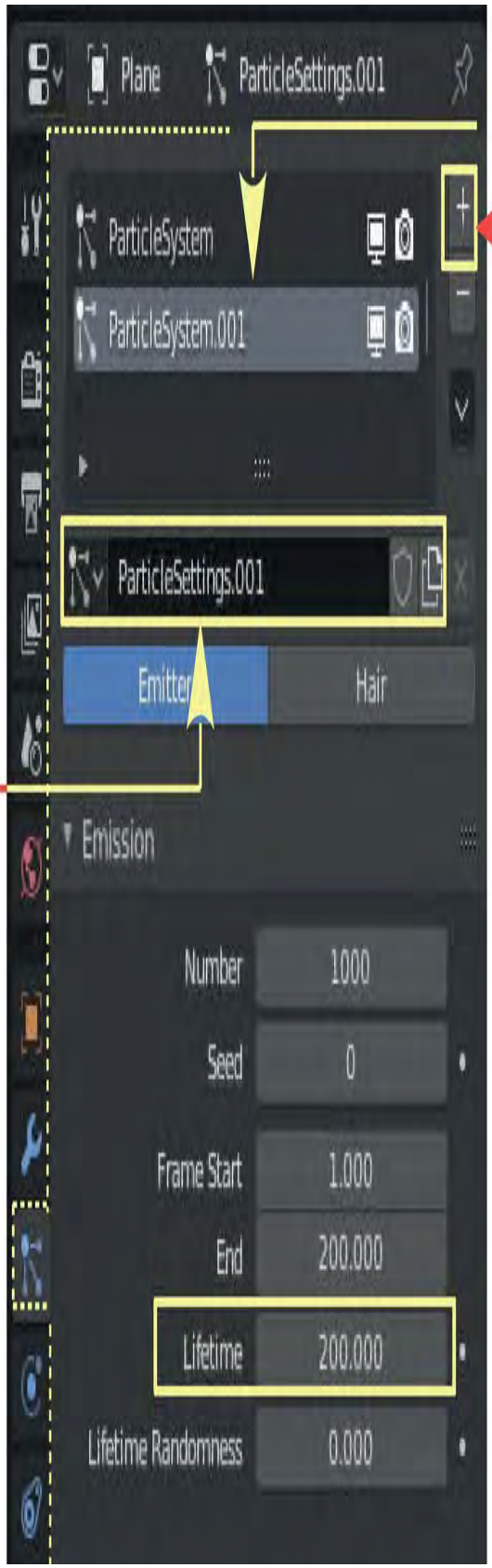
Figure 24.116

To add a second Particle System to the Plane click on the Plus icon adjacent to the Assignment Panel.

Change the **Lifetime** in the Emission Tab to 200 so that the Particles remain visible when the animation is played.

Add a **Monkey Object** to the Scene with a Material and park it to one side.

ParticleSystem.001 is entered in the Assignment Panel. **Note;** Settings: **ParticleSettings.001**.



Assignment Panel

Click to add a second Particle System

Figure 24.117

At this point a second Particle System is applied to the Plane with **Settings: ParticleSettings.001**. The Settings (at this point) are identical to the original ParticleSettings (Figure 24.117), therefore, playing the animation will produce two identical Particle displays (yellow spheres) and appear as if nothing has changed. To make it obvious that two systems are in play modify the settings for ParticleSettings.001.

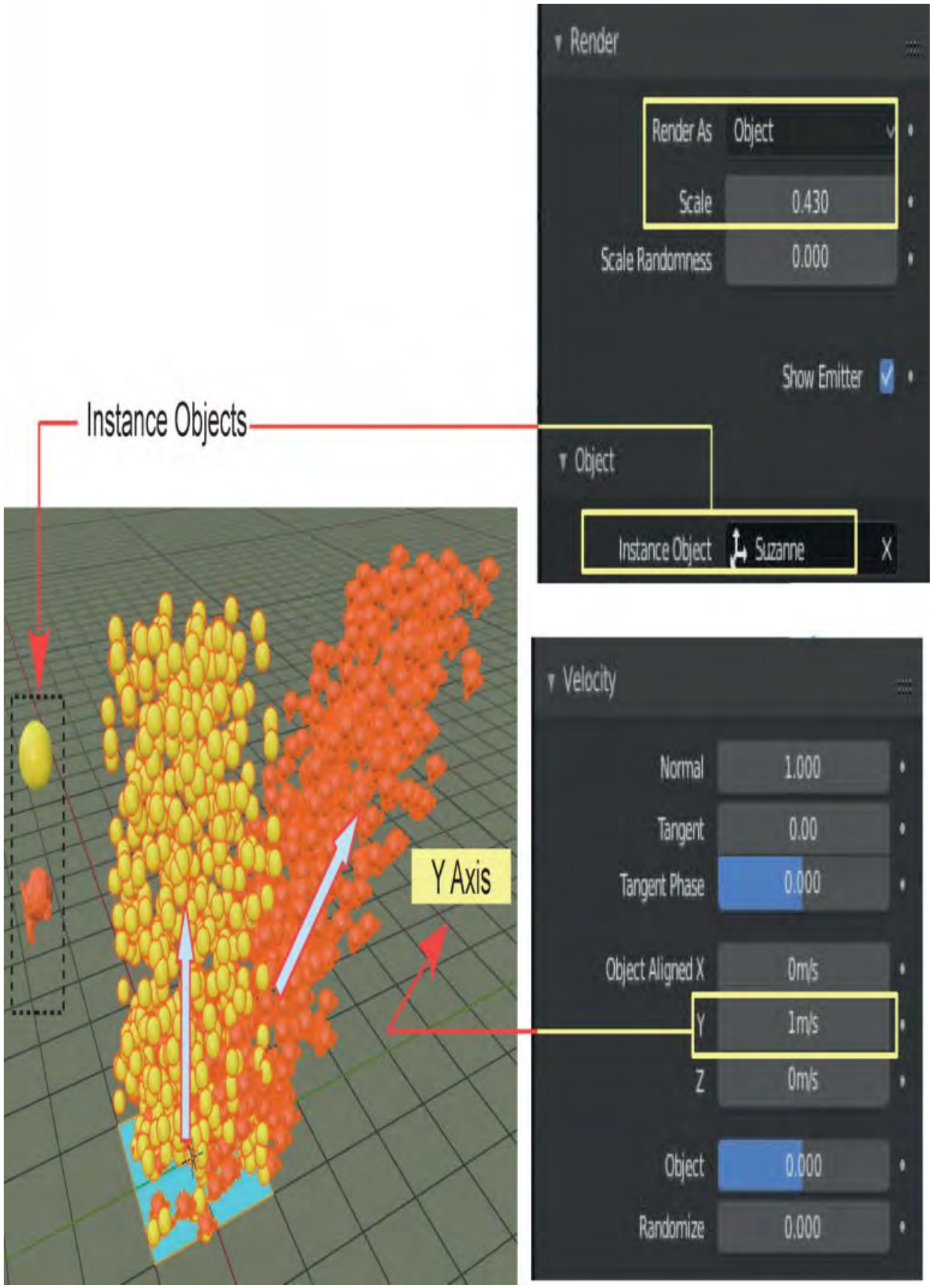

Render
Render As
Object
Scale
0.430
Scale Randomness
0.000
Show Emitter
Instance Objects
Object
Instance Object
Suzanne
X
Y Axis
Velocity
Normal
1.000
Tangent
0.00
Tangent Phase
0.000
Object Aligned X
0m/s
Y
1m/s
Z
0m/s
Object
0.000
Randomize
0.000

Figure 24.118

With **ParticleSettings.001** selected (highlighted) change the values in the **Render Tab** as shown in Figure 24.118.

Change values in the **Velocity Tab** (Figure 24.120) as shown to change the direction of emission of the second set. Play the animation.

24.23 Keyed Particles

Keyed Physics is a way of controlling the movement of particles by directing them from the original Emitter Object to a second Target Object and onto subsequent Target Objects. The flow of Particles may be used as an animation or used to create a static image. The following procedure for setting up a keyed system will demonstrate the principles involved.

Open a new Scene in Blender and delete the default Cube. Add three separate Plane Objects and position them as shown in Figure 24.121. Note that in the **Outliner Editor** under **Collection** the first Plane (red) will be named **Plane**. The second Plane (blue) named **Plane.001** and the third (green) **Plane.002**.

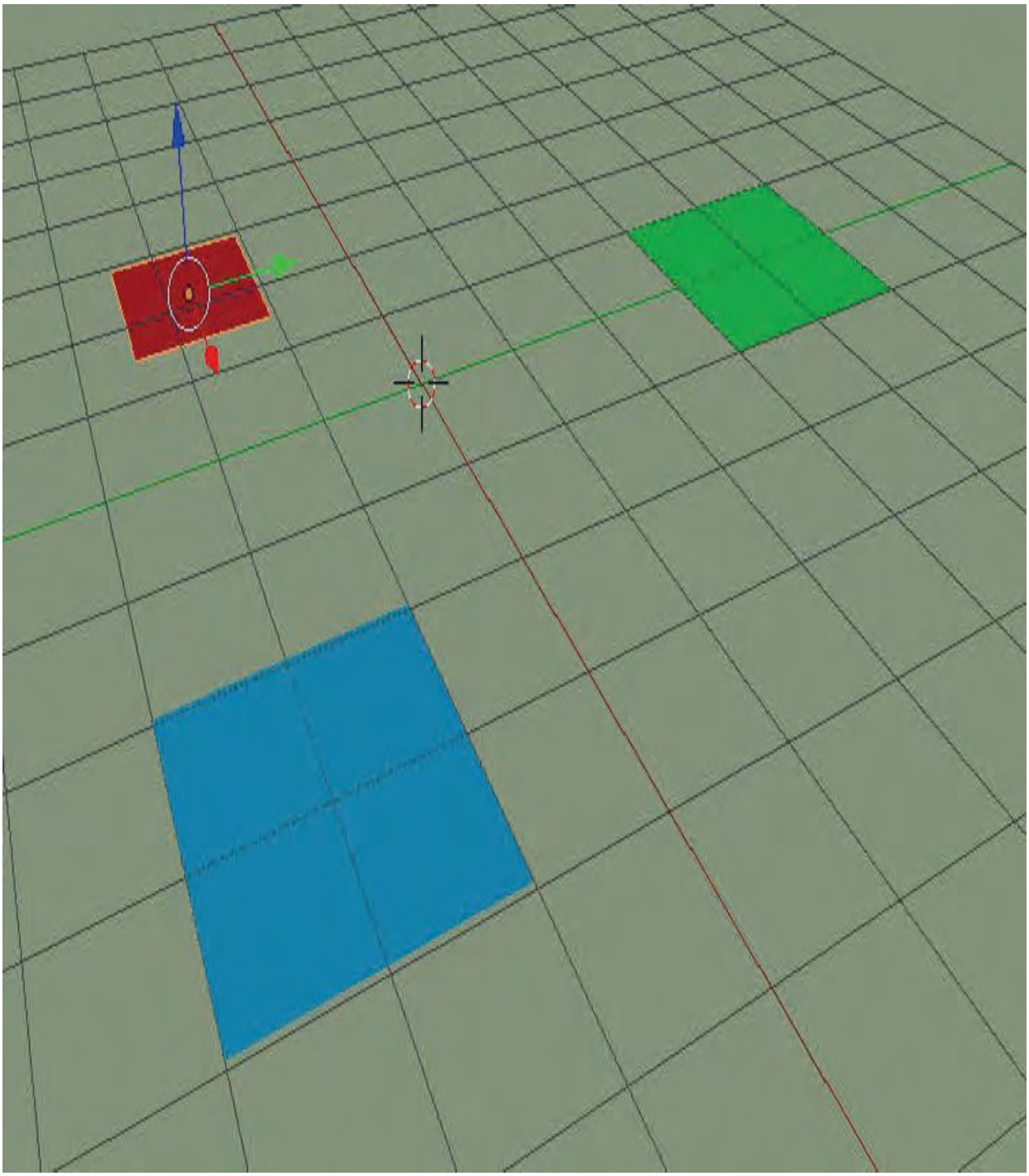


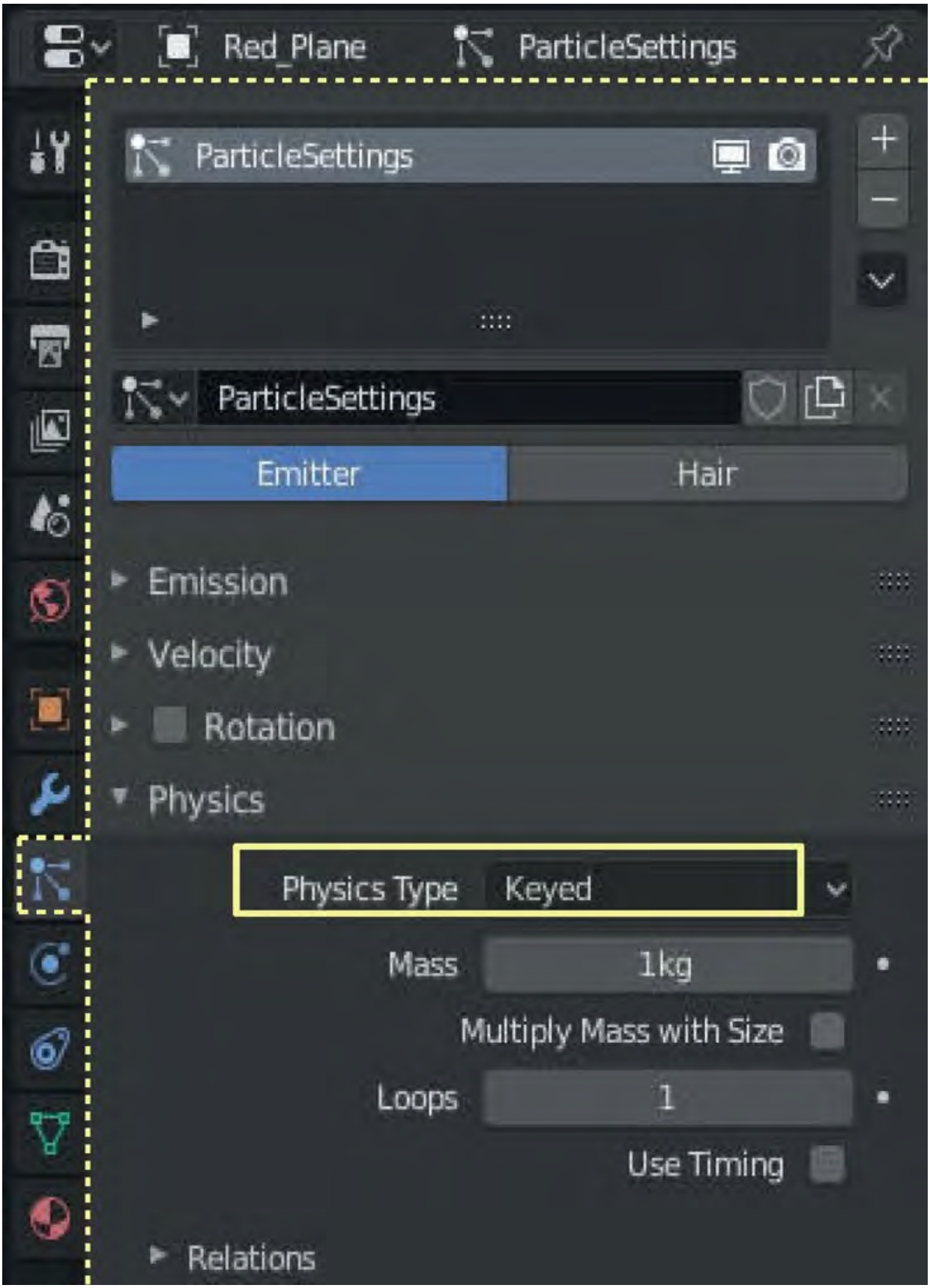
Figure 24.121

Figure 24.122

In the **Outliner Editor** rename the Planes, **Red_Plane, Green_Plane** and **Blue_Plane**.

Note: You can go to the Properties Editor – **Object Data Properties** and edit the name in the **Name box** at the top of the panel or you can edit the names in the **Outliner Editor**. You could use any name you like but since the Planes are colored it's best to name per the colors assigned.

Select the first plane, **Red_Plane** and in the **Properties Editor, Particles Properties,** add a Particle System. In the **Tab** Newtonian to **Keyed**.


Red_Plane
ParticleSettings
ParticleSettings
ParticleSettings
Emitter
Hair
Emission
Velocity
Rotation
Physics
Physics Type
Keyed
Mass
1kg
Multiply Mass with Size
Loops
1
Use Timing
Relations

Deselect **Red_Plane** in the 3D Viewport Editor and select **Green_Plane** and **Blue_Plane** in turn, repeating the procedure for adding a **Keyed** type particle system.

Go back and select **Red Plane**. At this point you are about to tell the Particles¯ emitted from Red_Plane to migrate to Blue_Plane then on to Green_Plane. This is done by designating Blue_Plane and Green_Plane as **Targets** in the **Physics, Keys, Relation Tab**.

Click on the plus sign at the RHS of the Relations panel. This enters a Target Channel and inserts **Invalid target**. Below the relations panel click in the **Target Object** panel and select **Red_Plane**.

With Red_Plane selected click the plus sign a second time and repeat the process, this time entering **Green_Plane** as the Target. Repeat for **Blue_Plane**.

OK! You have the objects, **Red_Plan, Green_Plane** and **Blue_Plane** each with a **Keyed** particle system and you have told the particles generated from **Red: Plane** to assemble.

Figure 24.124

When the animation is played Particles emitted from Red_Plane travel towards Blue_Plane then turn and head over to Green_Plane. To alter the direction, select one of the channels in the relations Panel then click a Up or Down Arrow to change the order.

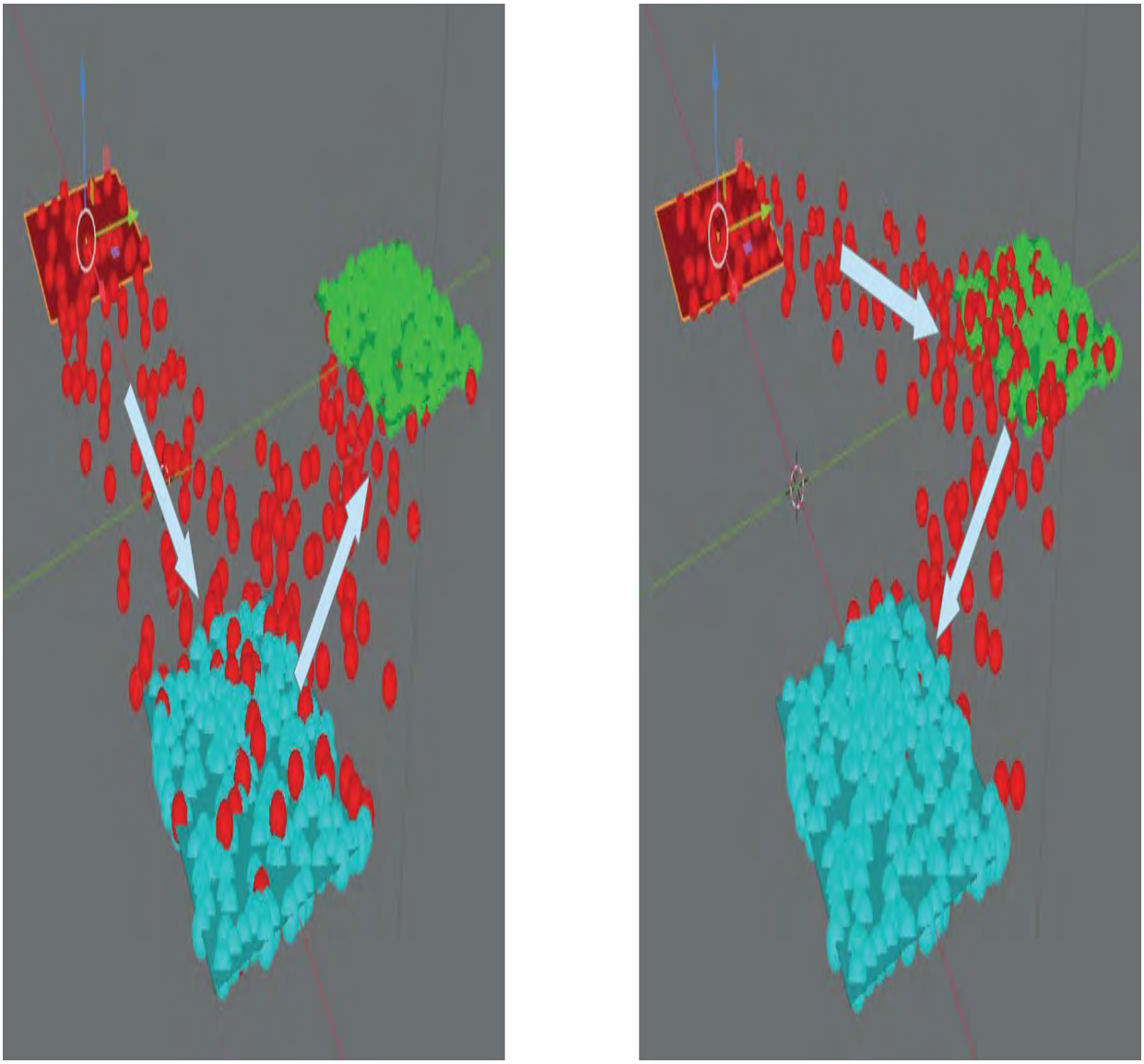


Figure 24.125

Remember: All the rules for number of particles, Lifetime, Start, End, and Normal velocity apply.

25 Physics and Simulation

25.1 Applying and Cancelling Physics
25.2 Real World Physics
25.3 Force Field
25.4 Collision Physics
25.5 Cloth Physics
25.6 Soft Body Physics
25.7 Fluid Simulation
25.8 Quick Methods
25.9 Fluid Simulation Continued
25.10 Fluid Simulation Experiment
25.11 Smoke and Fire Simulation
25.12 Smoke and Fire from Scratch

Simulation refers to Animation where dynamic events which take place in the real world are depicted (simulated) in an Animated Sequence. When physical actions occur in the real world they obey the laws of Physics.

Characters and Objects in a Computer Animation are made to move and interact as if they were complying with Real World Physics. Characters jump up and fall down obeying the law of gravity. They collide with and bounce off each other and with obstacles in the Scene. These actions may be exaggerated to create effect and, therefore, do not strictly adhere to the laws of physics, depending on the story being depicted. Blender incorporates Modifiers to simulate how Characters and Objects behave as if following these laws.

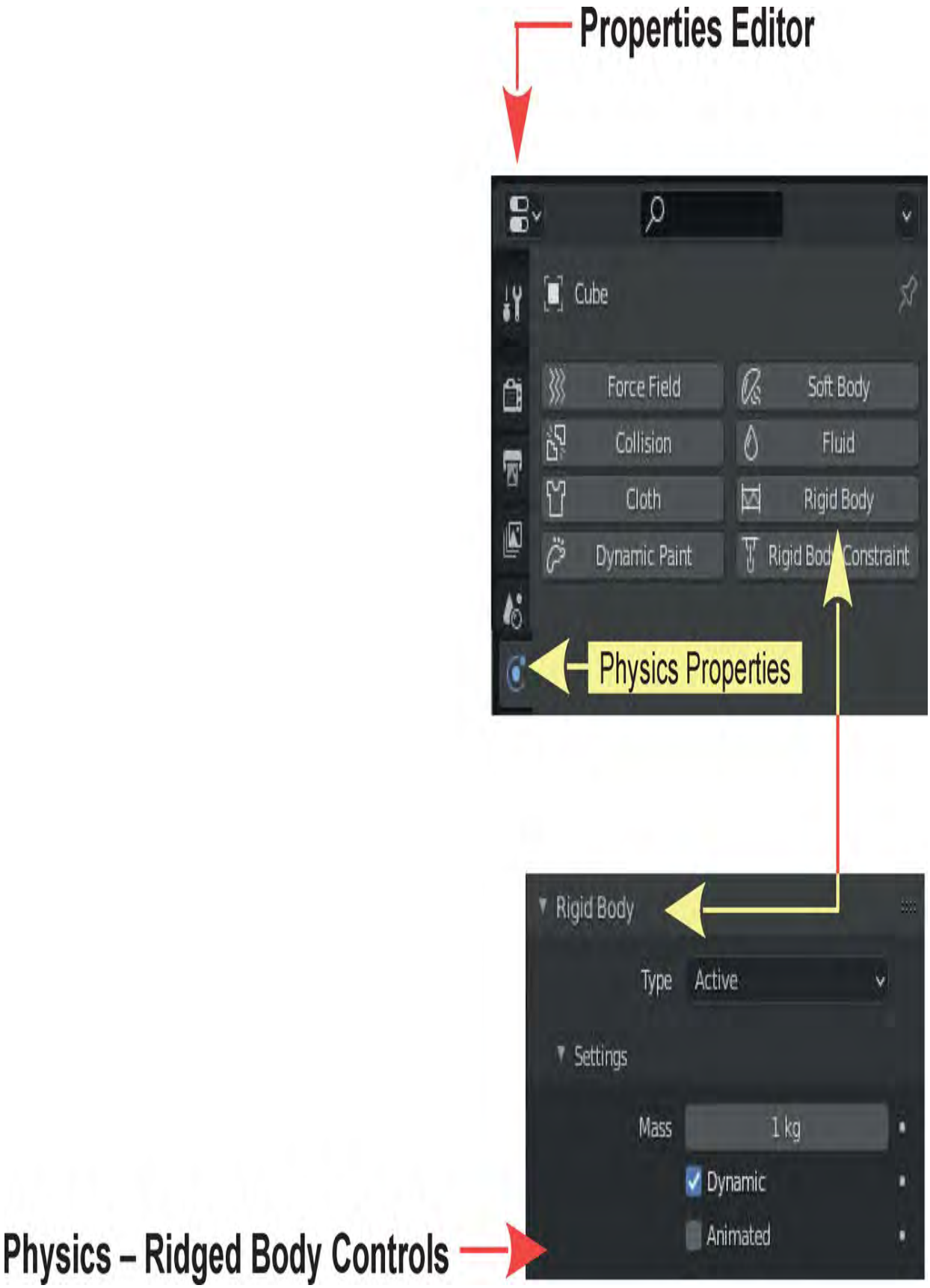

Properties Editor
Cube
Force Field
Soft Body
Collision
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Physics Properties
Rigid Body
Type
Active
Settings
Mass
1 kg
Dynamic
Animated
Physics – Ridged Body Controls

Figure 25.1
Figure 25.2

In the preceding chapter describing Particles and Particle Effects, Wind Force and Gravity have been encountered. When an Objects emits Particles they descend in the Scene as if under the influence of the real world force of **Gravity** and are affected by a **Wind Force** when applied. Gravitational Force is treated as a separate force with controls for cancelling the effect and applying the strength in the **Properties Editor, Scene Properties.** The Wind Force is found, with several other Physics simulation effects in the **Properties Editor, Physics Properties.**

Physics Properties are **Modifiers** (coding that produces an effect) and are applied to an Object which is selected in the 3D Viewport Editor, much the same as a Modifier is applied. When Physics is applied to an Object, controls for Physics display in the **Properties Editor, Physics Properties** (Figure 25.2).

25.1 Applying and Cancelling Physics

To apply Physics, select the Object in the 3D Viewport Editor then click on the **Physics Type** in the **Properties Editor, Physics Properties.** You will see that the icon preceding the Physics Type name changes to a cross indicating that Physics has been applied and controls for the Physics Type display in the Editor.

Should you wish to cancel the Physics for the Object, click on the cross.

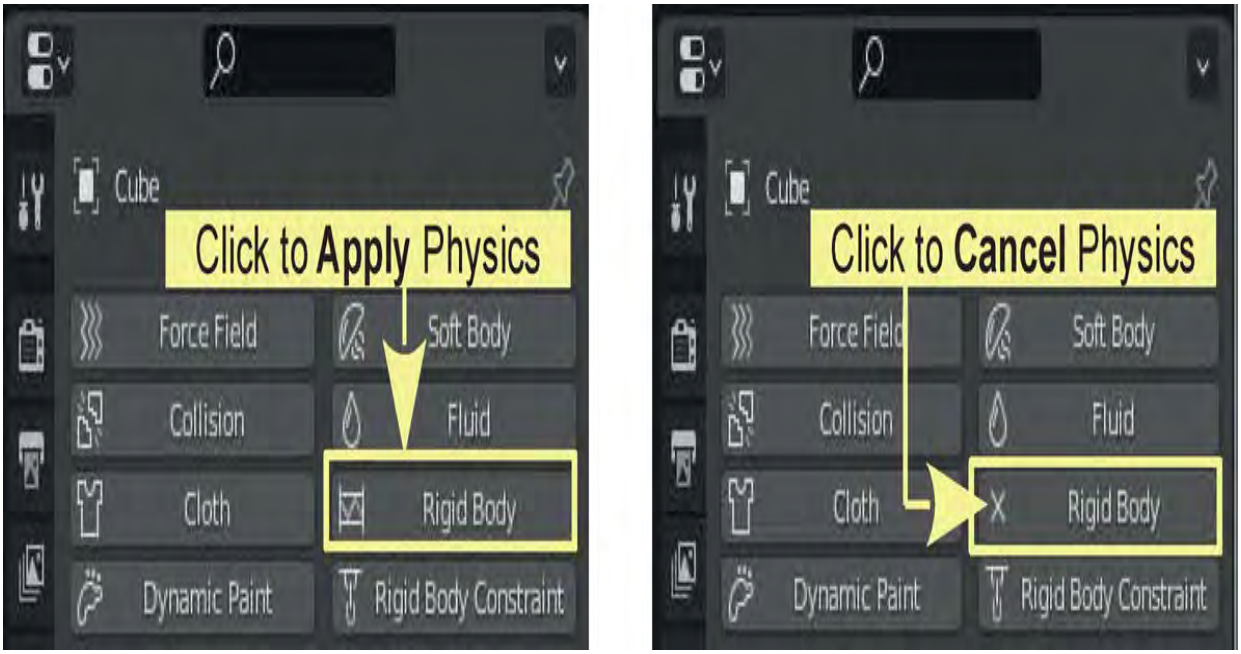


Figure 25.3

Physics may also be applied (in some cases) in the **Properties Editor, Modifier Properties.**

Note: Some Modifiers display in both the Physics Properties and the Modifier Properties.

Physics may also be applied (in some cases) in the **Properties Editor, Modifier Properties**.

Figure 25.4

Note: When applying a Physics Property from the Modifier Properties, a panel displays directing you to the Physics Properties for the controls.

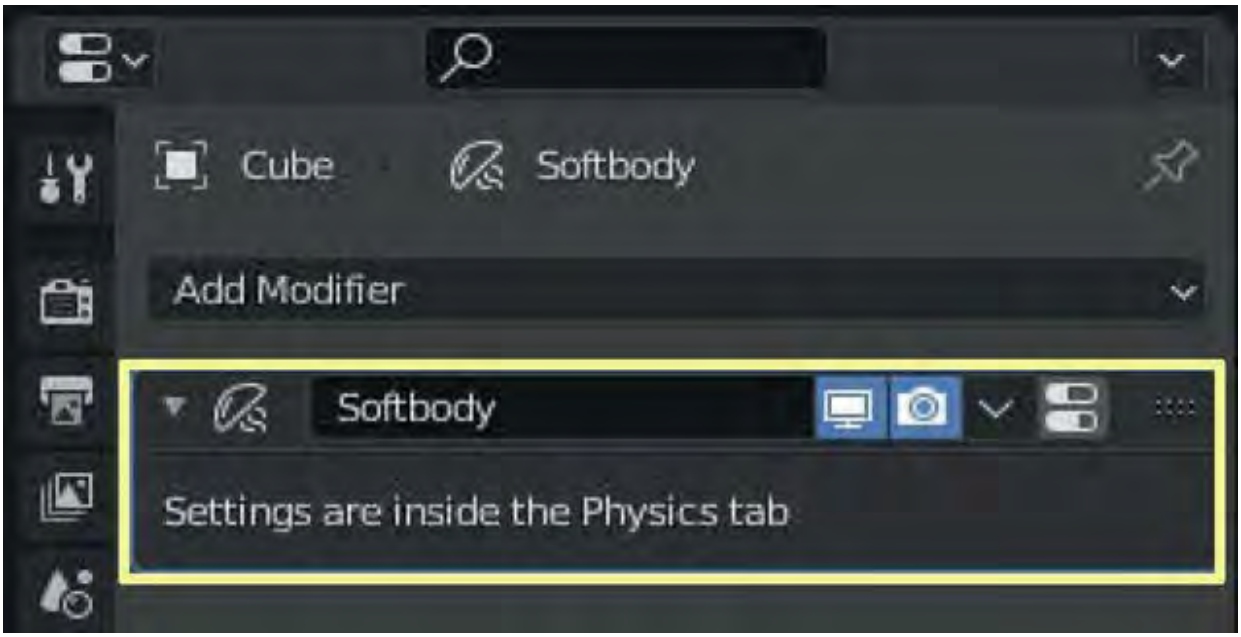


Figure 25.5

25.2 Real World Physics

In Chapter 22, you will have observed that Particles being generated from a default Object descended in the Screen when an animation sequence is played. The Emitter Object remains stationary unless it is animated to move.

The default Blender Scene may be considered to be a view of a 3D World. There is a gravitational force in existence, but generally, Objects placed in the Scene will not react to Gravity until **Physics** is applied to the Object. Particles are an exception since they have **Newtonian Physics** applied by default (Figure 25.6).

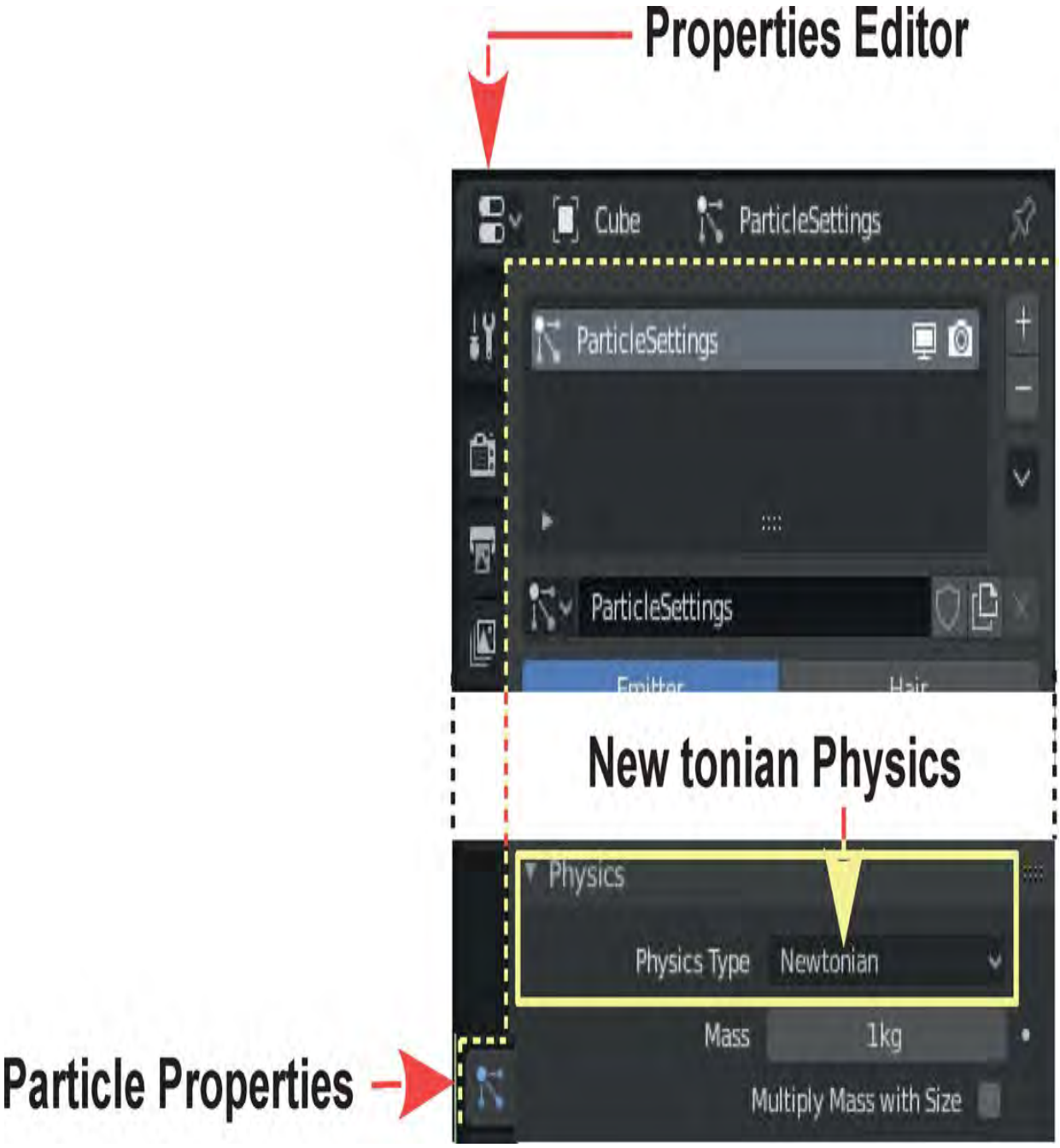


Figure 25.6

Real World Physics is referred to as; **Rigid Body Physics** (the Object is said to be a Rigid Figure 25.7 shows a UV Sphere positioned above a Plane which has been rotated on the X Axis to form an incline. The Sphere has a Particle System applied.

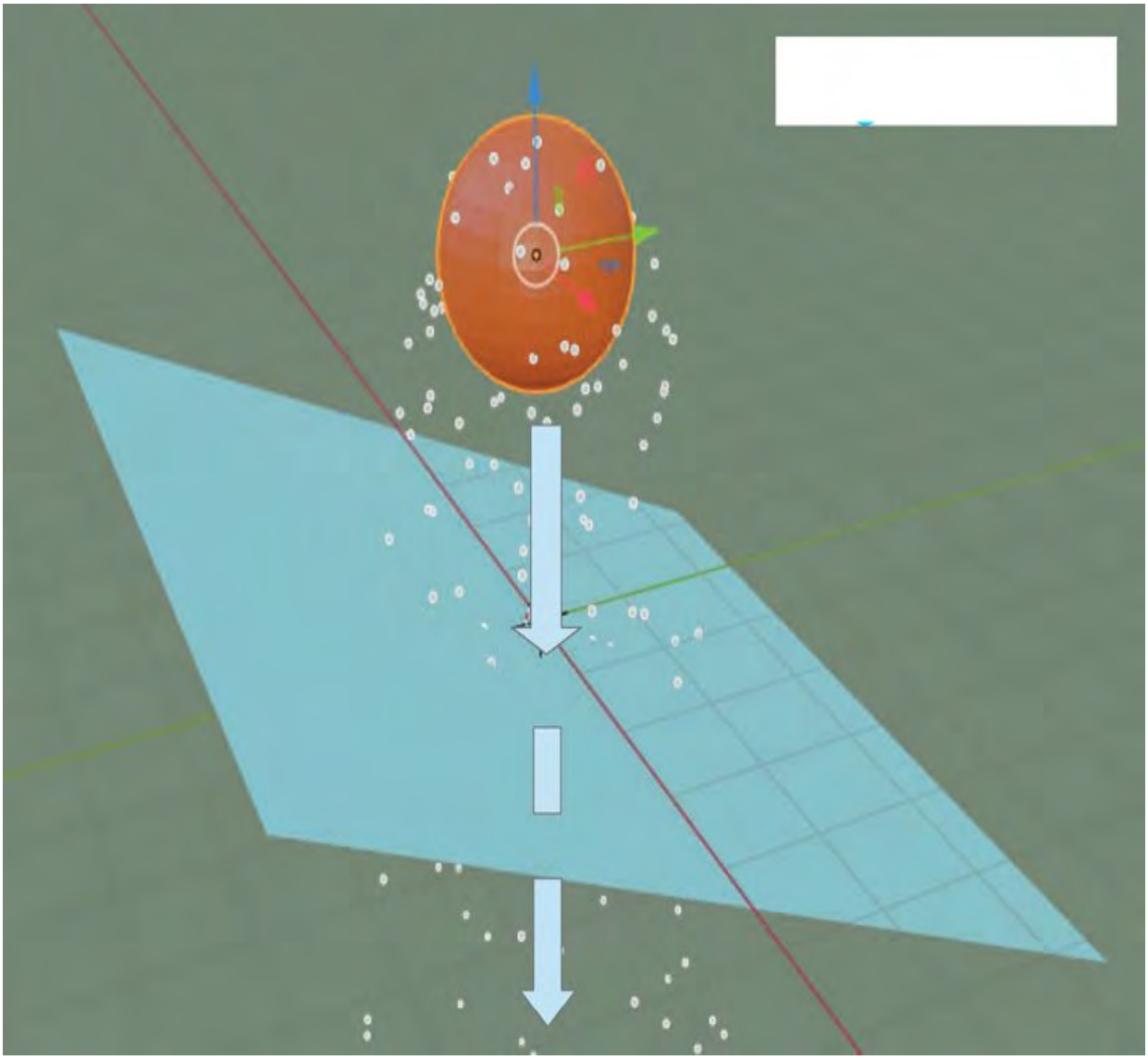


Figure 25.7

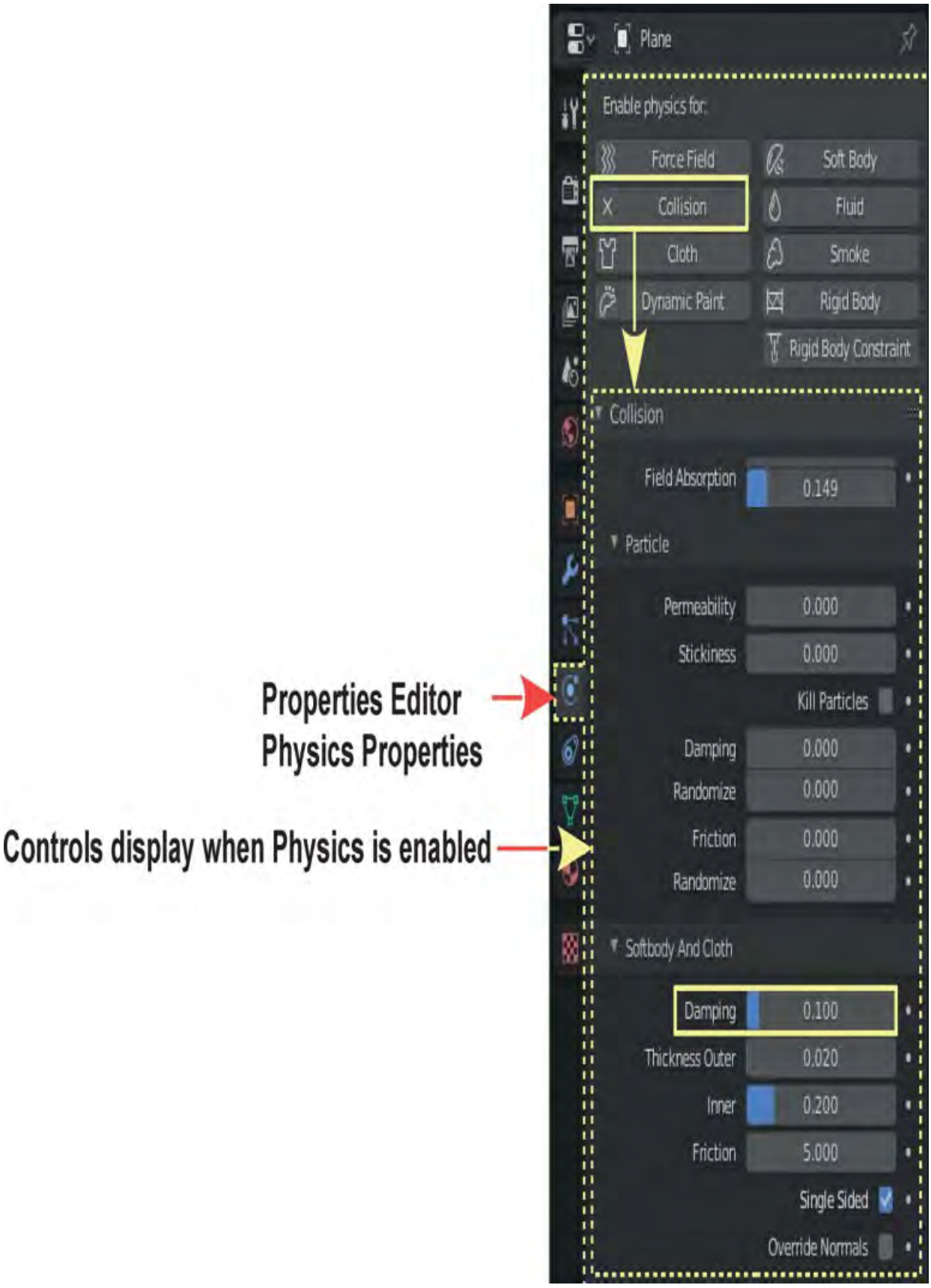

Properties Editor
Physics Properties
Controls display when Physics is enabled
Plane
Enable physics for:
Force Field
Soft Body
Collision
Fluid
Cloth
Smoke
Dynamic Paint
Rigid Body
Rigid Body Constraint
Collision
Field Absorption
0.149
Particle
Permeability
0.000
Stickiness
0.000
Kill Particles
Damping
0.000
Randomize
0.000
Friction
0.000
Randomize
0.000
Softbody And Cloth
Damping
0.100
Thickness Outer
0.020
Inner
0.200
Friction
5.000
Single Sided
Override Normals

Figure 25.8

When an animation is played in the Timeline Editor, Particles are Emitted from the Sphere and fall through the Plane. The Sphere and Plane remain stationary (Figure 25.7).

By selecting the **Plane** and **Enabling Physics** for: **Collision,** in the Properties Editor Physics Properties, then adjusting the **Damping** value in the Collision, Softbody And Cloth Tab (Figure 25.8) the Particles will bounce off the Plane when the animation is replayed (Figure 25.9).

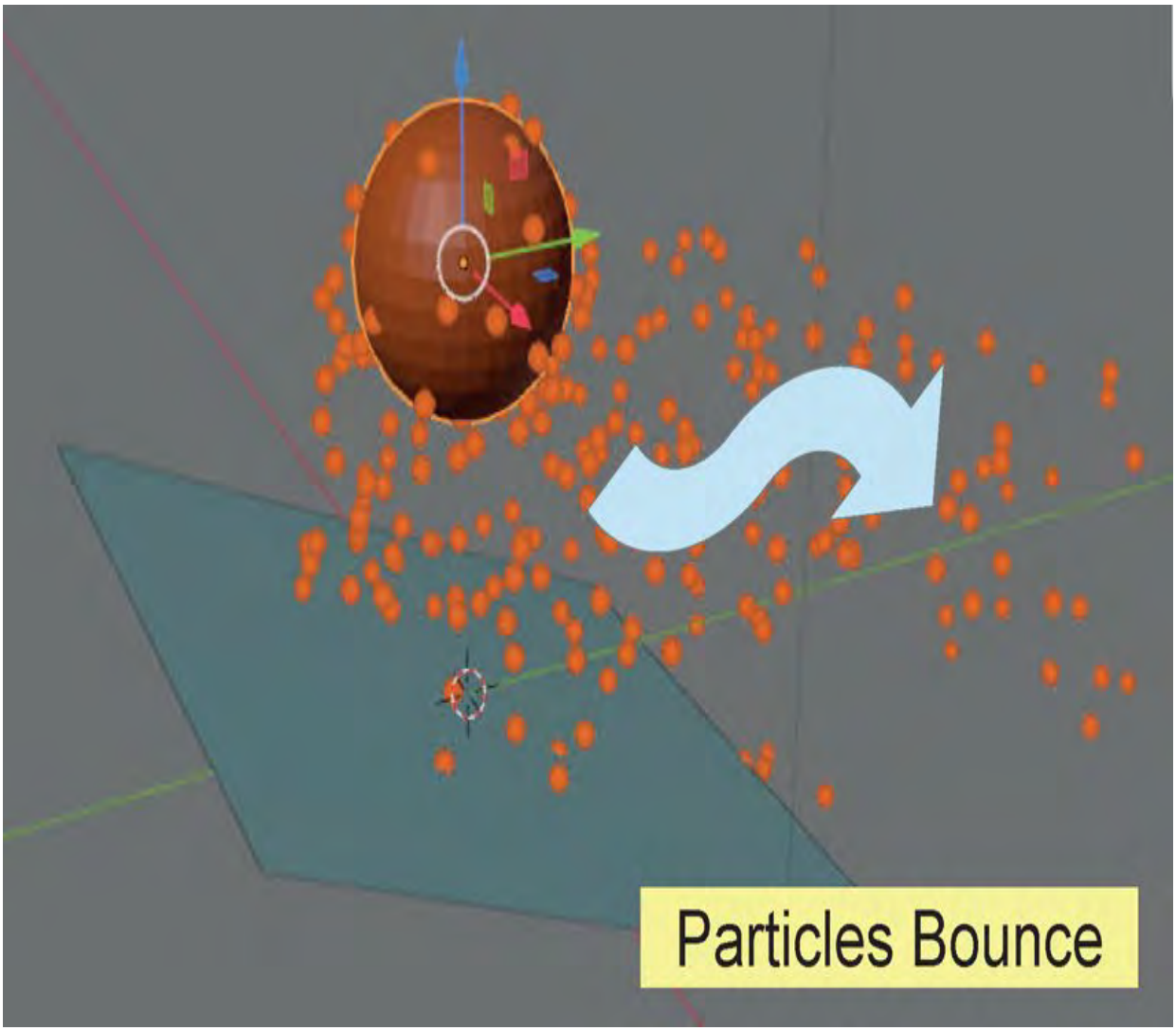


Figure 25.9

To have the Sphere fall and roll down the Plane, add **Ridged Body Physics** to both the Sphere and the Plane **AND** in the **Rigid Body Tab** set **Type: Active** for the Sphere (Figure 25.10) and **Type: Passive** for the Plane (Figure 25.11) . The Sphere will fall, the Plane remains stationary.

UV Sphere – Rigid Body Physics

Figure 25.10

Plane – Rigid Body Physics

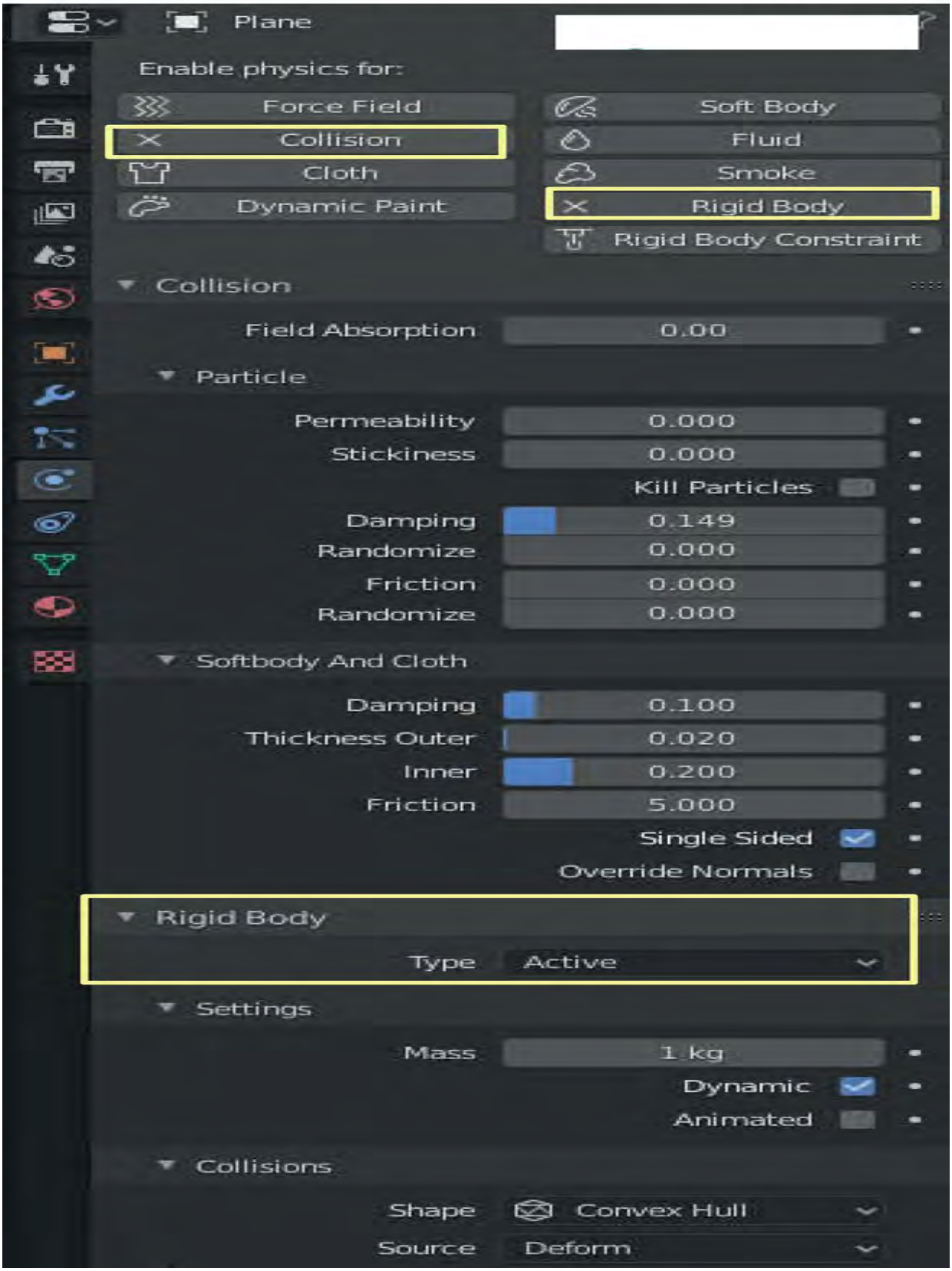


Figure 25.11

The Sphere falls, rolls down the Plane and over the edge. Particles follow (Figure 25.12).

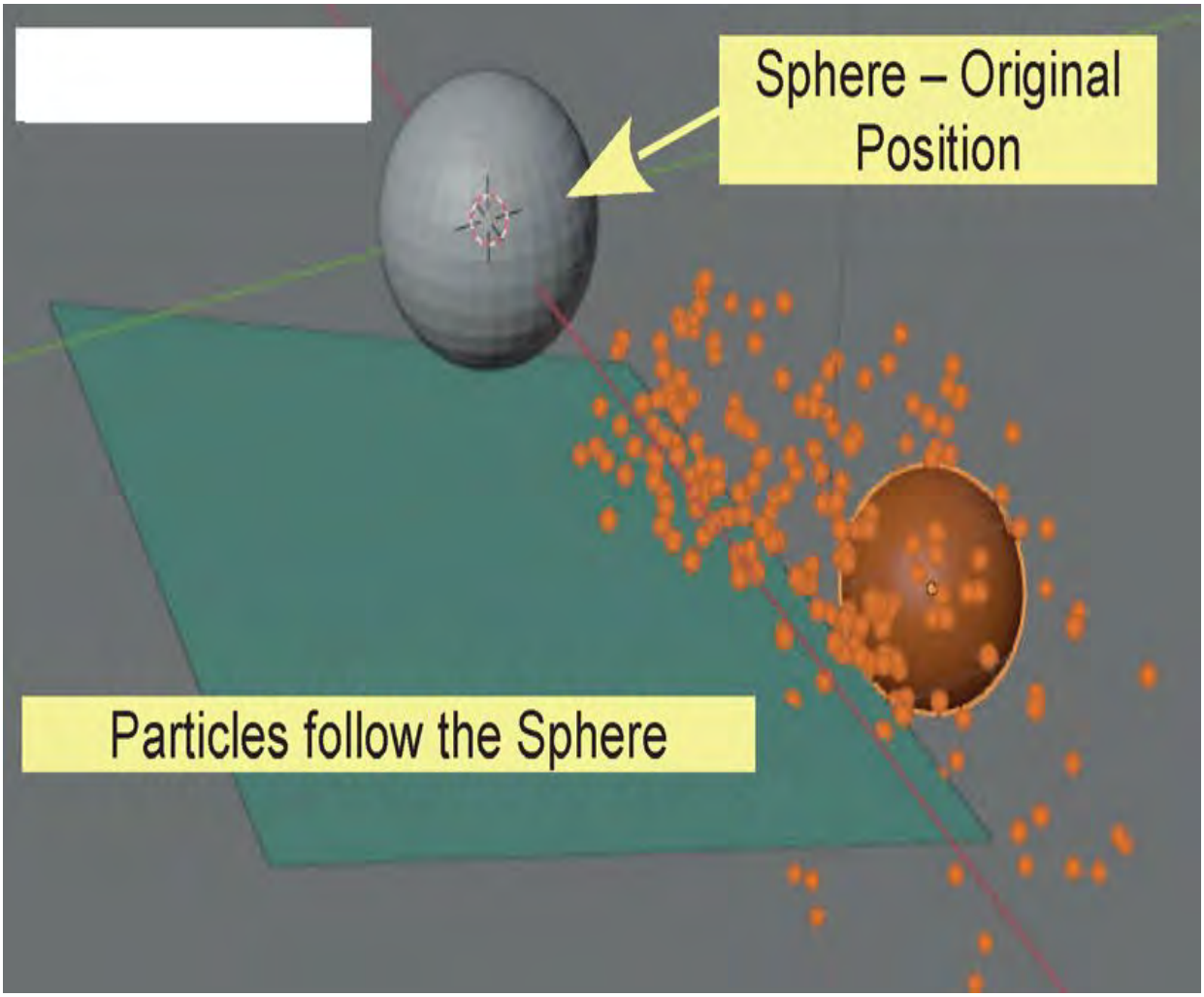


Figure 25.12

If you remove the **Collision Physics** from the Plane the Sphere still rolls down the incline on impact but the Particles being Emitted pass through the Plane (Figure 25.13).

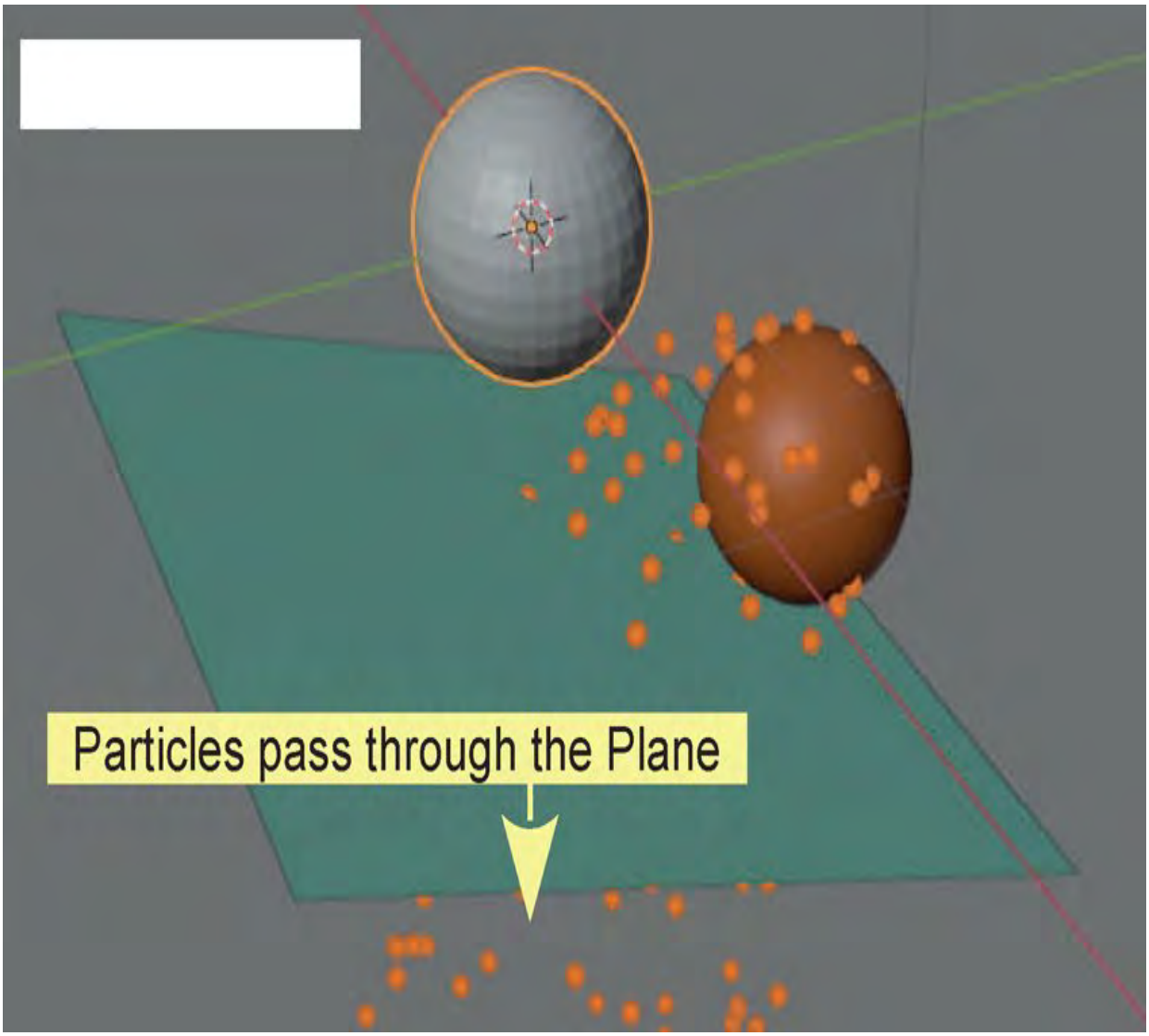


Figure 25.13

25.3 Force Fiele

An example of a Force Field was encountered in Chapter 24 – 24.14 where a Wind Force was used to influence the direction of flow of Particles. In that instance the Wind Force was assigned to an Empty Object in the Scene which enabled the Force to be positioned for effect.

A Wind Force may be applied to any Object. The Empty Object was used since it doesn't Render in the Scene. Figure 25.14 shows a Force Field –

Type Wind applied to a Cone Object which is directed to blow the Particles being emitted from the UV Sphere used in the previous exercise.

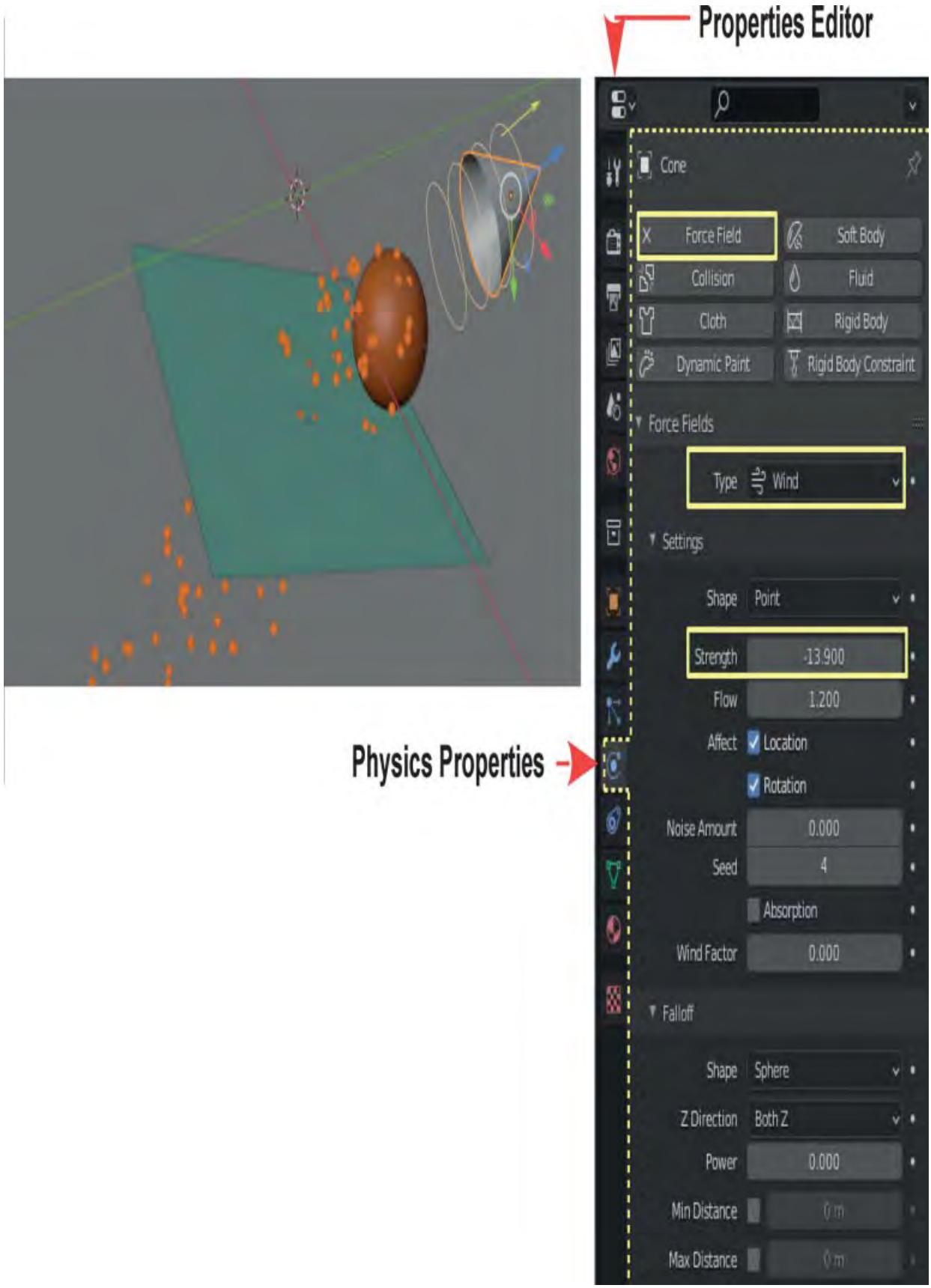

Properties Editor
Physics Properties
Cone
Force Field
Soft Body
Collision
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Force Fields
Type
Wind
Settings
Shape
Point
Strength
-13.900
Flow
1.200
Affect
Location
Rotation
Noise Amount
0.000
Seed
4
Absorption
Wind Factor
0.000
Falloff
Shape
Sphere
Z Direction
Both Z
Power
0.000
Min Distance
0 m
Max Distance
0 m

Figure 25.14

The Particles being emitted from the Sphere as it falls and rolls to the right down the Plane are blown to the left.

Note: the negative **Strength** value for the Wind Force.

25.4 Collision Physics

As seen when discussing Real World Physics Section 25.2, Applying **Collision Physics** to an Object causes it to interact with other Objects in the Scene.

25.5 Cloth Physics

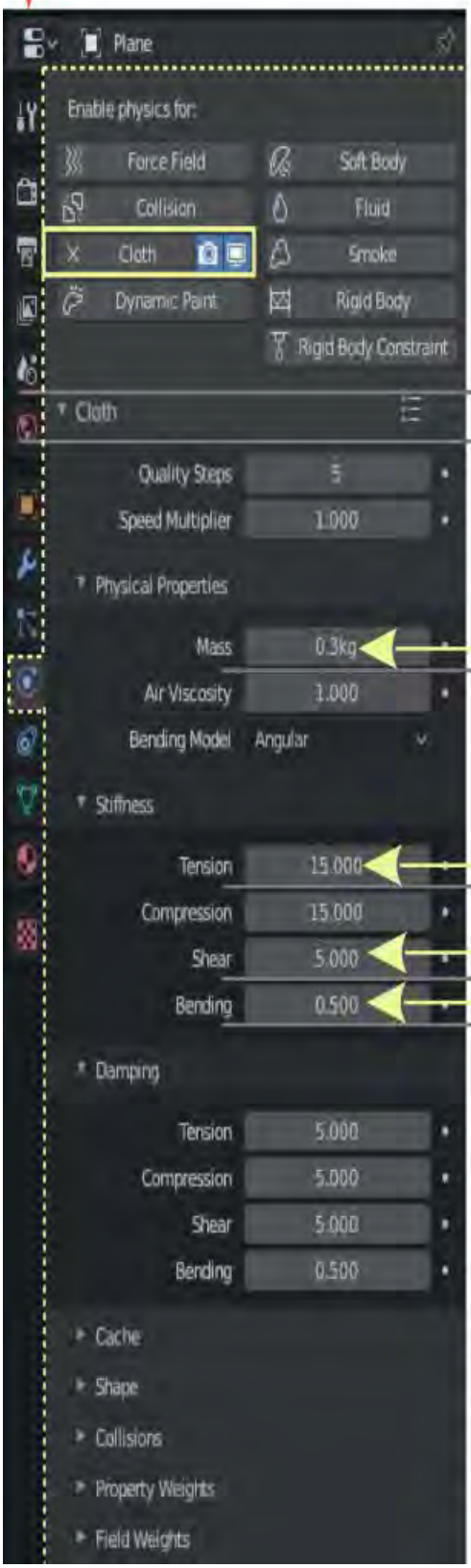
With **Cloth Physics** applied, an Object exhibits the characteristics of different types of fabric. The controls for Cloth Physics display in the Properties Editor (Figure 25.15).

Consider a **Plane Object** in the 3D Viewport Editor in Object Mode, **Subdivided ten times** (in Edit Mode). With the Plane selected in **Object Mode,** click **Cloth** in the **Physics Properties.** If you click on the Modifier Properties in the Properties Editor you will see that a Cloth Modifier has been added to the Plane referring you to the controls in the Physics Properties.

Physics Cloth Properties

The default Cloth settings in the Physics buttons give the Plane the characteristics of a cotton fabric.

Properties Editor – Physics Properties



The default Cloth settings in the Physics buttons give the Plane the characteristics of a cotton fabric.

As a guide the following setting changes may be made for other **fabric materials**.

	Default				
	Cotton	Leather	Rubber	Denim	Silk
Mass	0.300	0.4	3.000	1.000	0.150
Tension	15	80	15	40	5
Shear	5	25	25	25	0.0
Bending	0.5	150	25	10	0.05

To see the Plane acting as a cotton fabric leave the default settings in place.

At this point playing an animation in the Timeline Editor will see the Plane exhibit the characteristics of a piece of cloth as if it had been released in space after being laid out perfectly flat. The cloth simply falls away under the influence of Gravity. There is no air resistance or other obstacle to impede its descent.

Figure 25.15

Cloth interacts with and is affected by other moving objects, the wind and other forces, as well as a general aerodynamic model, all of which is under your control.

Cloth example.

Cloth on carved wooden men (made by motorsep).

Cloth example.

Figure 25.16

To see the Plane acting as a cotton fabric leave the default settings in place.

At this point playing an animation in the Timeline Editor will see the Plane exhibit the characteristics of a piece of cloth as if it had been released in space after being laid out perfectly flat. The cloth simply falls away under the influence of Gravity. There is no air resistance or other obstacle to impede its descent.

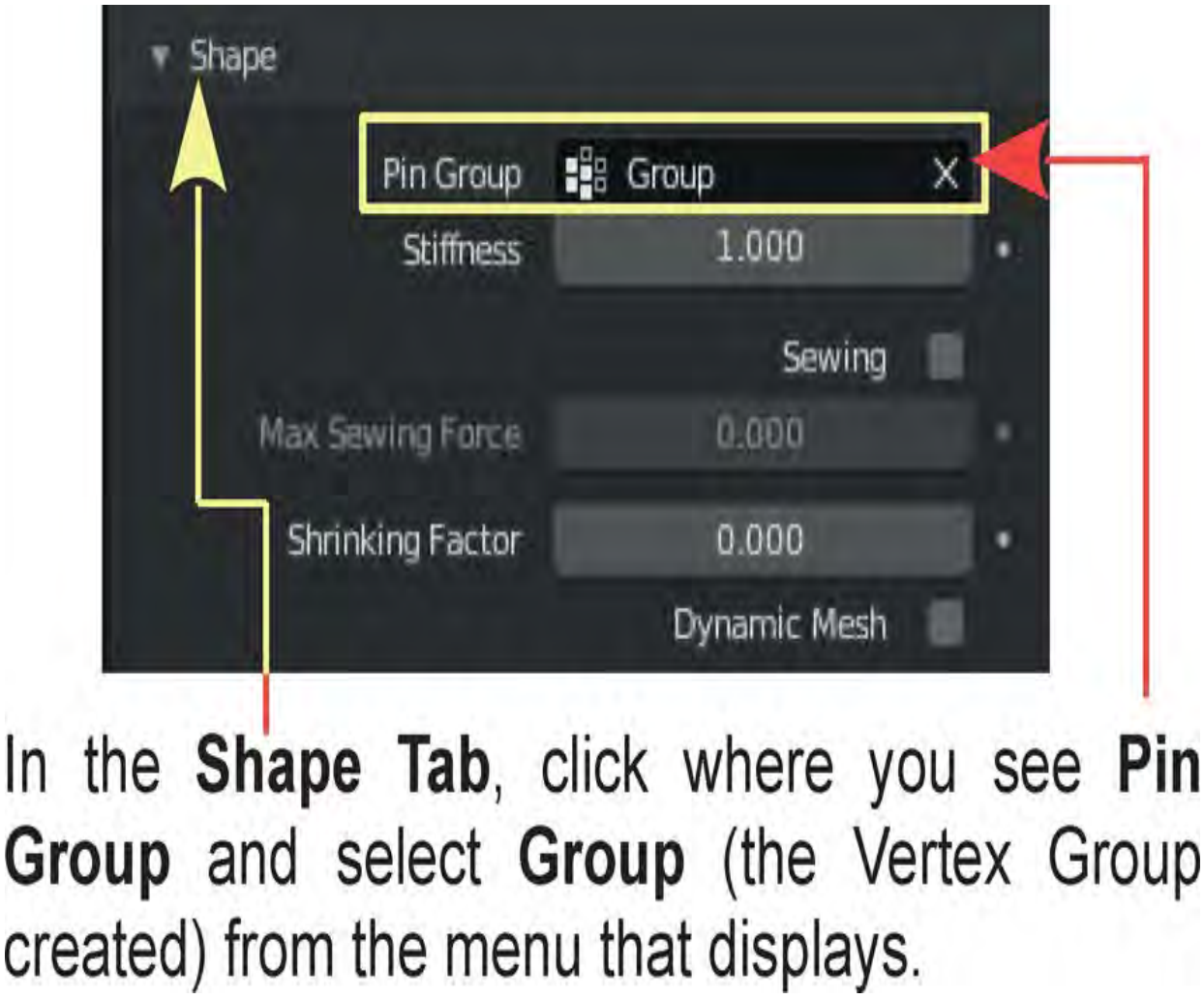


Figure 25.17

With the Plane selected in the 3D Viewport Editor, Tab into Edit Mode, deselect all vertices then select two corner vertices and create a **Vertex Group** (Chapter 5 – 5.9) with just the two. The Vertex Group is names **Group. In** the **Properties Editor, Physics Properties** expand the **Shape Tab** (way down the bottom Figure 25.17). You are about to fix (Pin) the two Vertices in space, like the corners of a sheet on a clothes line, **without the line.**

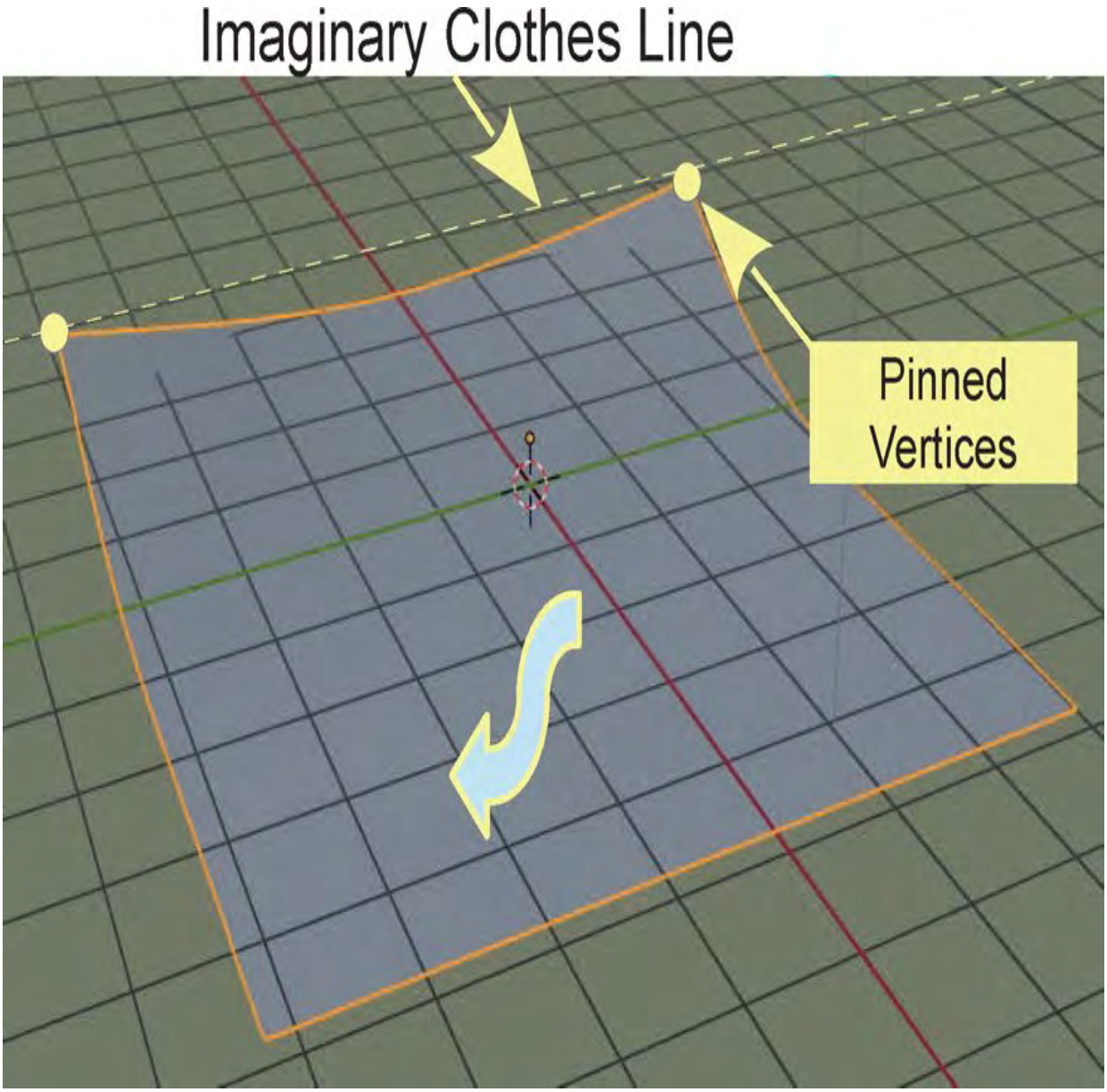


Figure 25.18

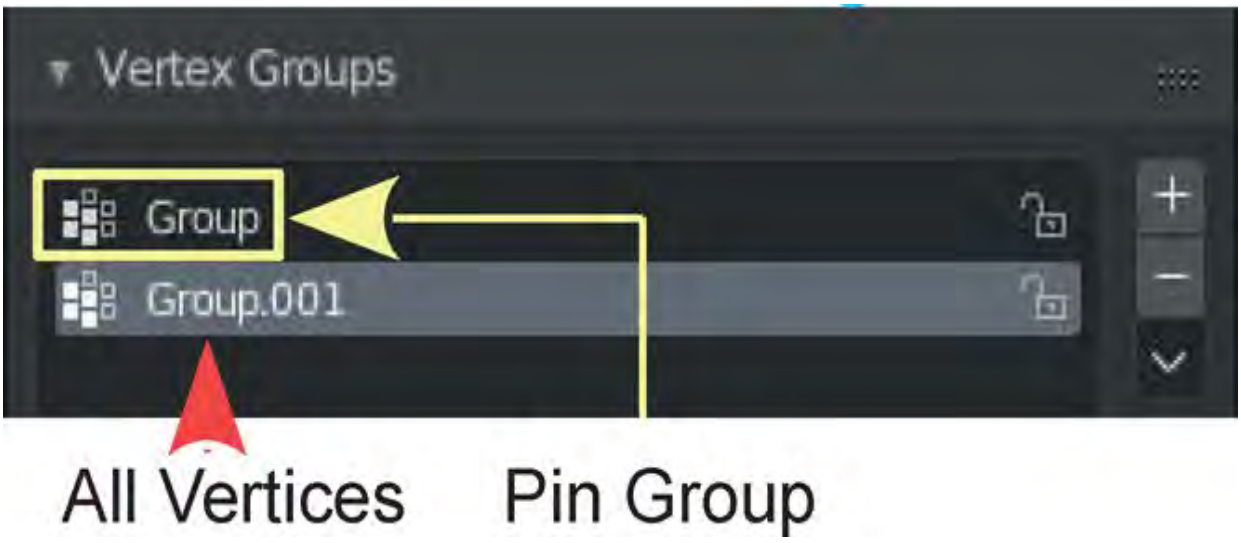


Figure 25.19

Tab back to Object Mode. Replay the animation in the Timeline. The Cloth swings down pinned at the corners (Figure 25.18).

To further demonstrate Cloth Physics, position a UV Sphere below the Plane **before playing the animation.** Create a second Vertex Group for the Plane, this time, including all Vertices on the Plane. **Activate Collision in the Physics Properties for the Sphere.**

Properties Editor, Physics Buttons

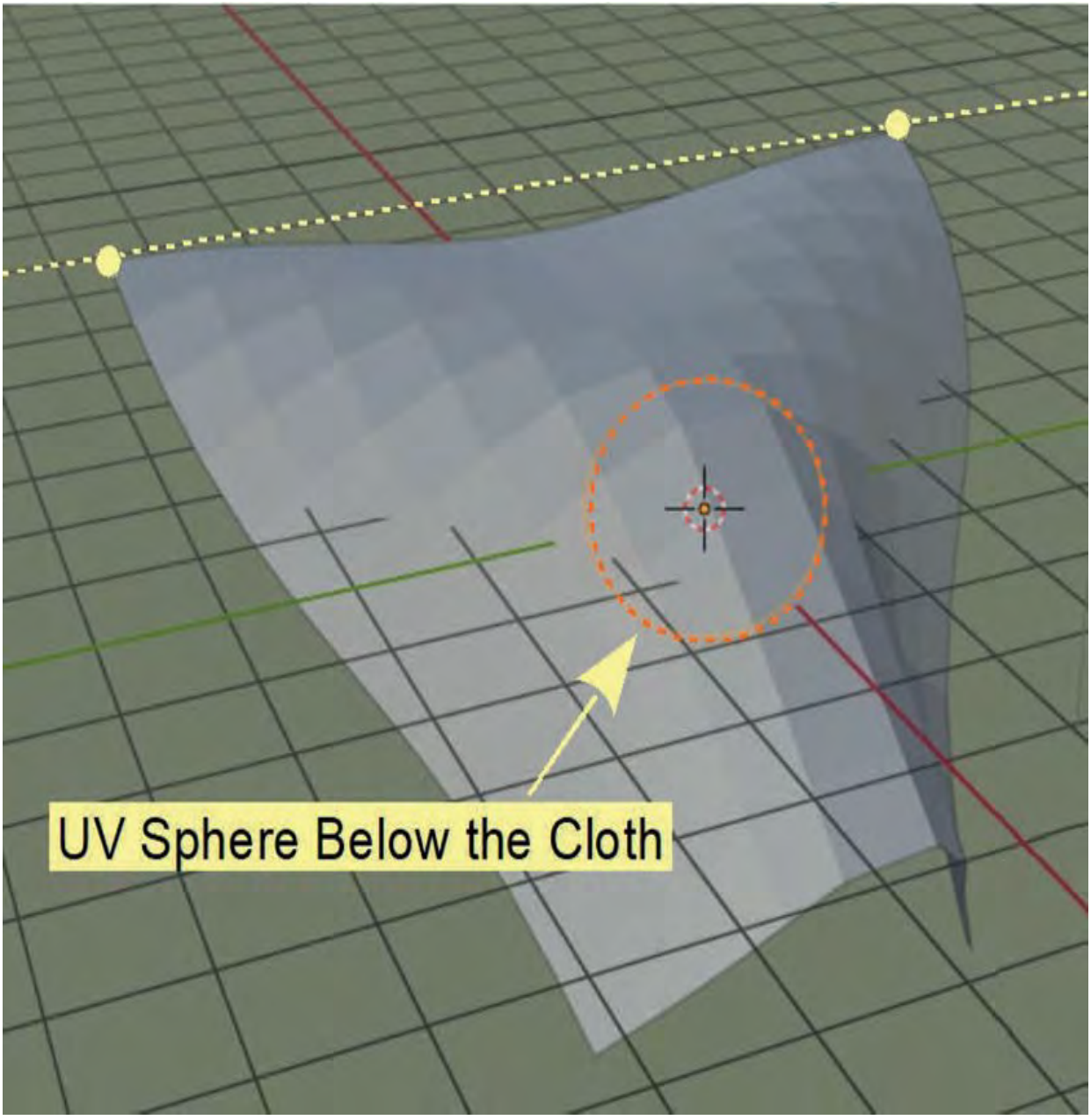


Figure 25.21

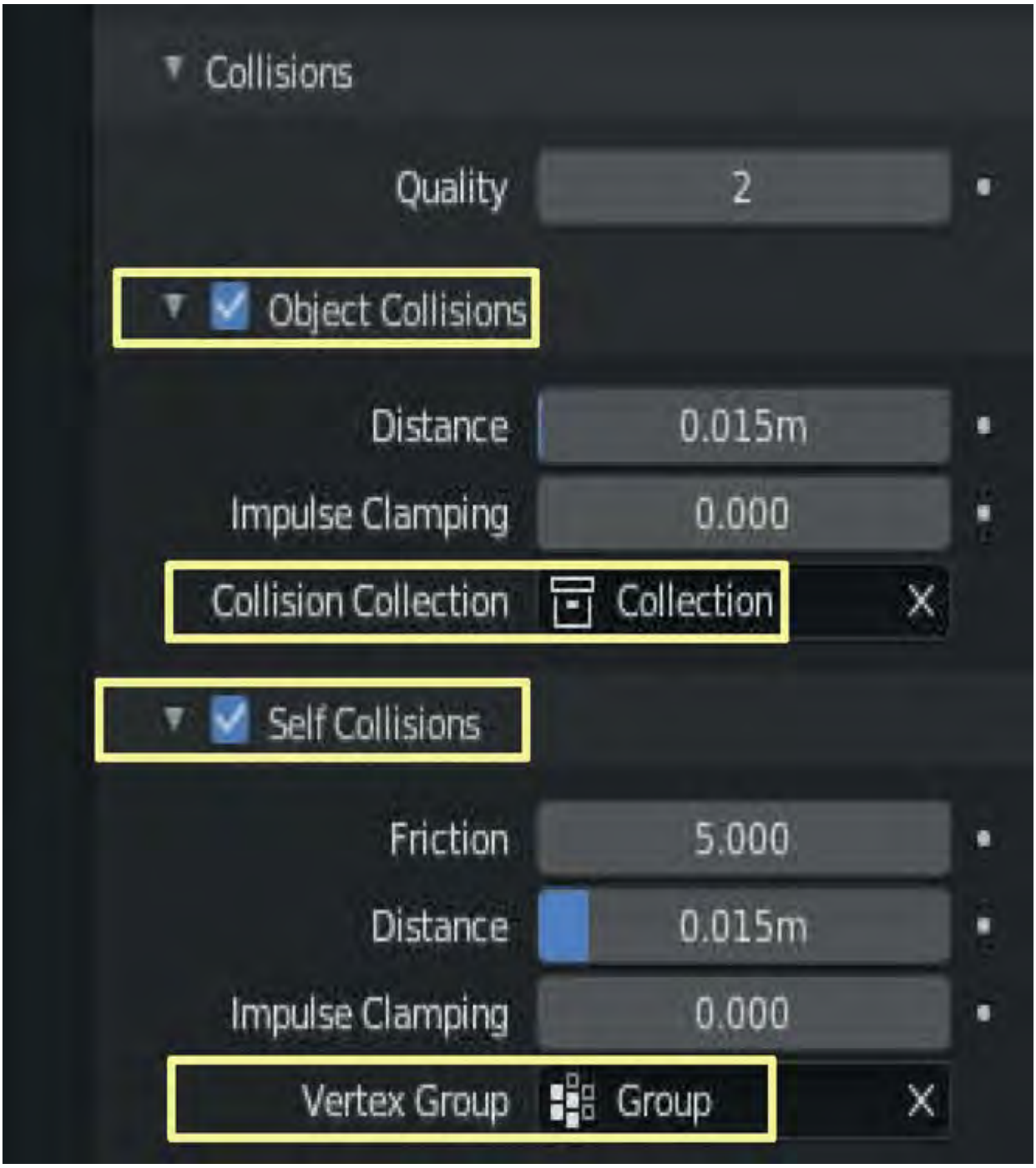


Figure 25.20

With the Plane selected, in the **Cloth Physics Properties, Collisions Tab,** have **Object Collision** checked with **Collision Collection: Collection** entered. Also have **Self Collision** checked with **Vertex Group: Group**

entered (Figure 25.20). Replay the animation to see the Cloth droop over the Sphere (Figure 25.21).

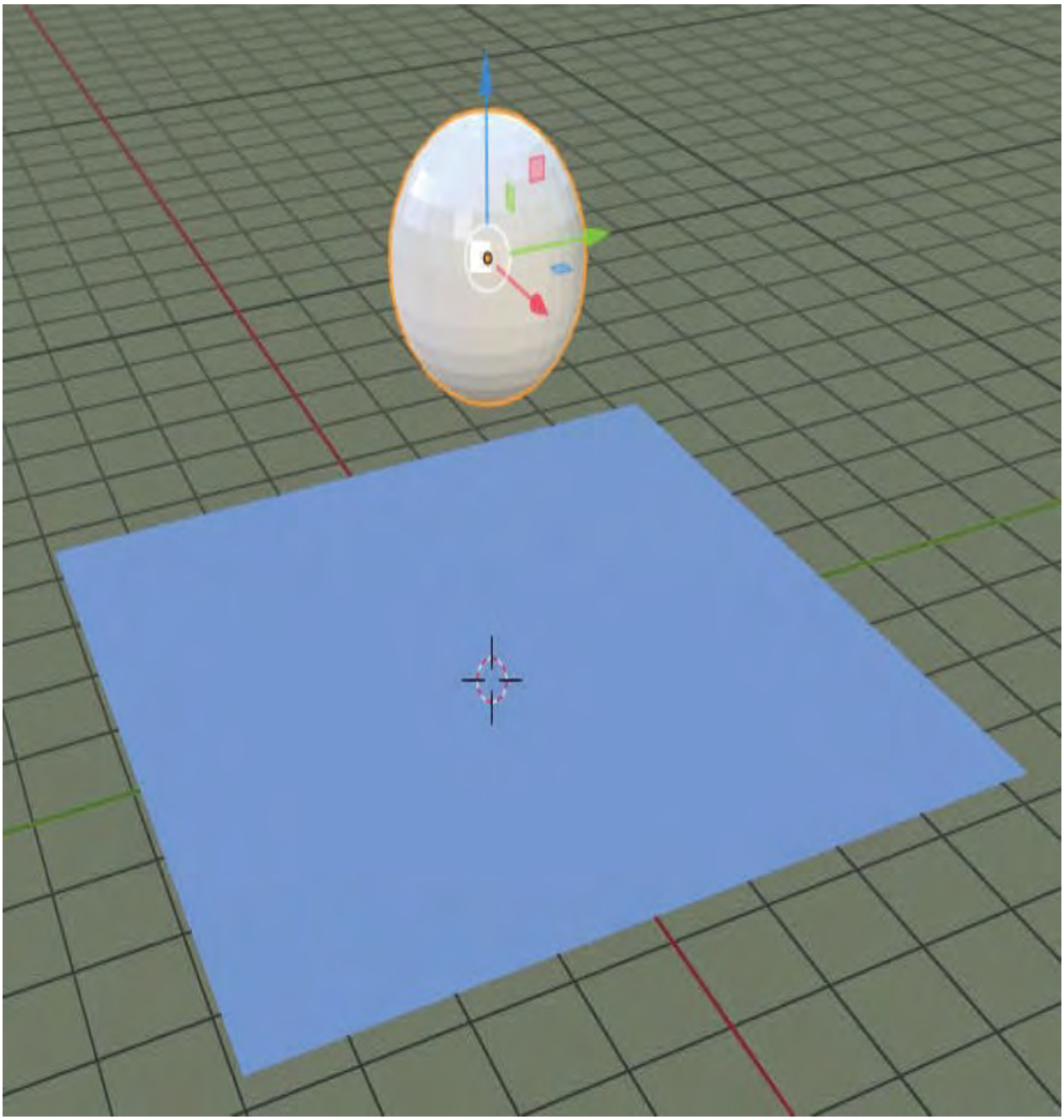


Figure 25.22

Note: Dynamic Paint is the next entry in the Physics list. This is detailed in Chapter 26.

25.6 Soft Body Physics

Soft Body Physics, when applied to an Object, causes it to act like dough or clay or anything that is soft and pliable.

As an example, set up a UV Sphere Object above a Plane (Figure 25.22). The Sphere is a good object for the demonstration since it has a reasonable number of Vertices forming its mesh surface.

With the Sphere in Object Mode add **Soft Body Physics** to the UV Sphere.

Playing an animation in the Timeline Editor at this point sees the UV Sphere remain in situ and bounce up and down slightly.

In the **Properties Editor, Physics Properties, Soft Body Tab, uncheck Goal** (Figure 25.23) to have the Sphere fall in the animation. The Sphere falls straight through the Plane.

Select the Plane and in the Properties editor, Physics buttons, **enable Physics for: Collision.**

The speed and deformation of the Sphere on impact may be controlled by adjusting values in the **Edges Tab** with the Sphere selected.

Figure 25.23

Playing the animation again, shows the UV Sphere crumpling when it makes contact with the Plane. Initially the crumpling will be hesitant and the animation will hesitate due to the time for the computer to calculate the deformation. Let the animation run its course. The deforming data will be written to the Cache (memory) then on replay the Sphere will squish into a blob and settle on the Plane. The amount of hesitation in the initial animation will depend on how fast the computer can perform the calculation.

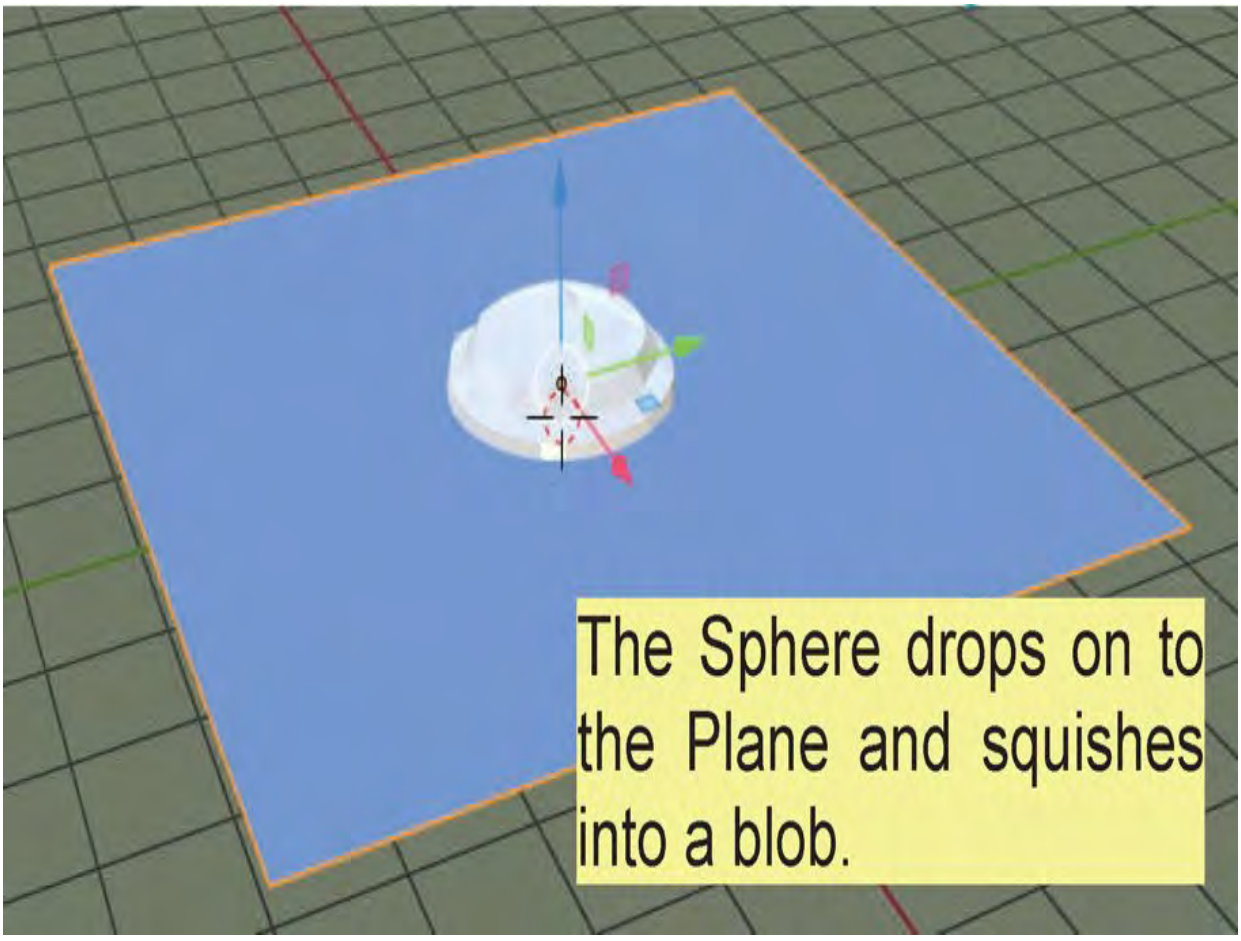


Figure 25.24

25.7 Fluid Simulation

Fluid Simulation, replicates how fluid behaves in the real world. How a fluid behaves depends on its environment which comprises physical obstacles and physical forces such as gravity and pressure. The physical composition of the fluid also has an effect on its behaviour. As you can guess, the laws of physics have a great deal to do with how a fluid behaves.

In Blender, a fluid simulation means, a graphical display is generated on the computer screen tricking the observer into believing they are seeing a fluid react to an environment. It should be remembered that this is an illusion and what you think you see is a clever bit of trickery.

In Blender an extensible framework called **Mantaflow** has been incorporated in the program for the execution of fluid and gas simulations.

Mantaflow is an open-source framework targeted at fluid simulation research in Computer Graphics. Its parallelized C++ solver core, python scene definition interface and plugin system allow for quickly prototyping and testing new algorithms. Mantaflow serves as the simulation engine in Blender. This brief definition means its a clever bit of gear for creating gas and liquid simulations.

The Concept in Blender

In a Fluid Simulation, the Fluid you see on the computer Screen, is initially represented by Particles. The Fluid is generated from a series of Particle Systems which represent the Fluid itself and may incorporate, Spray, Foam

and Bubbles. The Particles are eventually Rendered as Animated Objects which have Materials applied and incorporating Scene Lighting produce a realistic Fluid Flow.

In a nutshell, the procedure for creating a Fluid Simulation is to create a space in the Scene in which the Fluid Flow it to take place (the Domain), insert an Object in the Domain to act as a Fluid Inflow Source and to insert other Objects which act as obstacles and controls for directing the flow. The Fluid Inflow Source is a Particle Emitter Object.

To demonstrate the basic concept of Fluid Simulation work through the following, not so simple, exercise beginning with the default Blender Scene containing the Cube Object.

Place the 3D Viewport Editor in **Wireframe Display Mode.**

Domain and Emitter Set Up

In the default Blender Scene, Scale the default Cube up twice on the Z Axis. Deselect the Cube and add a second Cube scaling it down in size (S Key + 0.173). Both Cubes will be located at the center of the 3D World. Move the second, smaller Cube, up towards the top of the default Cube (Figure 25.25). The default Cube will be the **Domain** (the mini artificial World) and the second smaller Cube will be the **Fluid Emitter.** Note; it is very important to keep all Objects participating in the Fluid Simulation entirely inside the Domain (nothing protruding outside) (Figure 25.25).

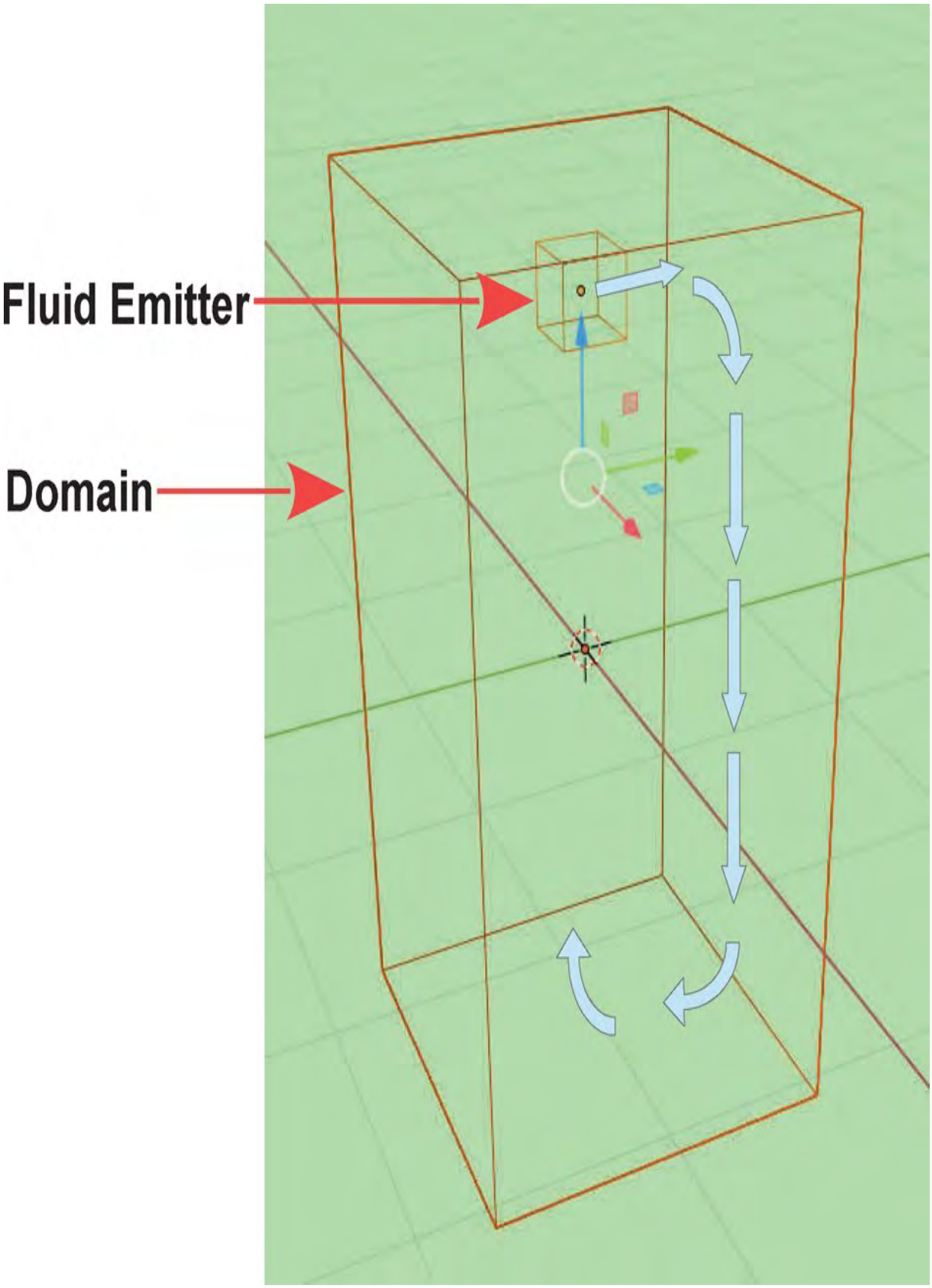

Fluid Emitter
Domain

Figure 25.25

When setting up a **Domain** containing a Fluid Emitter, the scale of the Emitter relative to the Domain is important. A large Emitter will produce a large amount of fluid in a short time.

In the **Outliner Editor** rename the smallCube, **Fluid** and the default Cube,**Domain** (Figure 25.26).

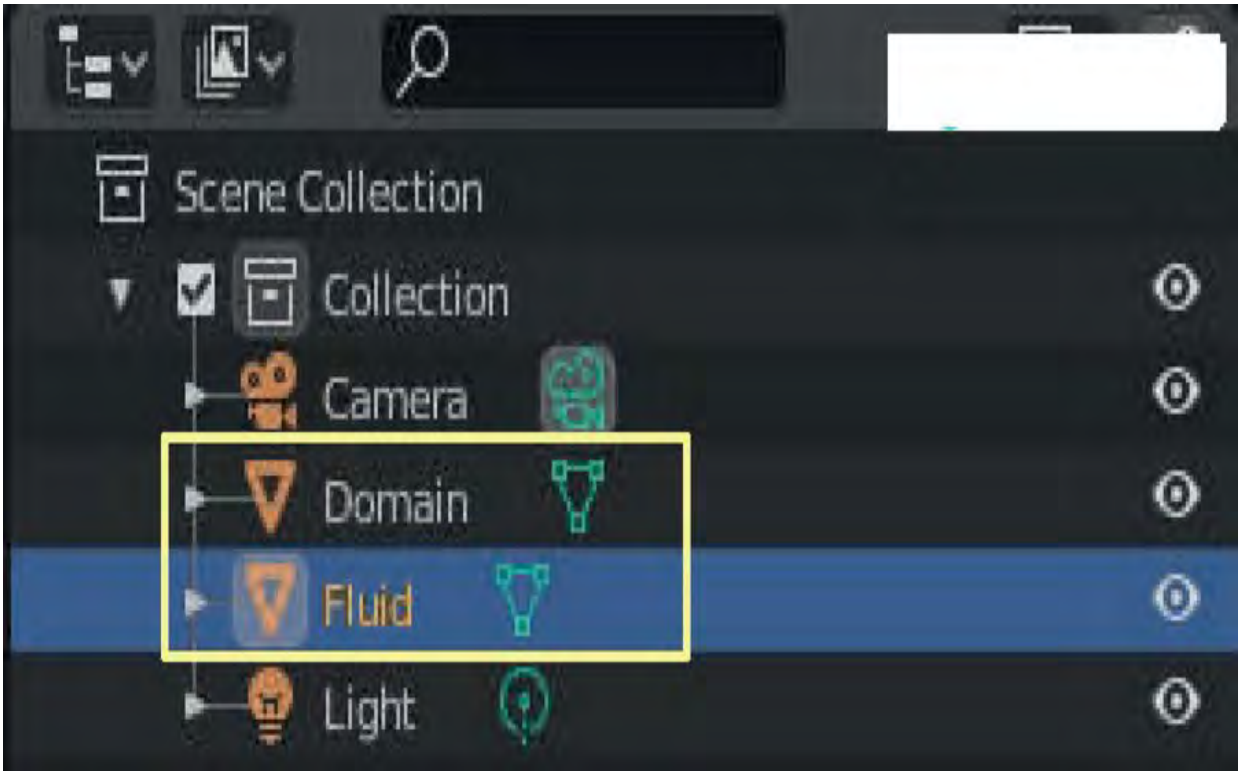


Figure 25.26

Fluid Flow

In this simple demonstration Fluid will be generated by the Emitter Object and delivered as a continuous stream. The stream will be, initially, directed along the Y Axis where it will contact the side of the Domain before descending and filling the Domain as if it were a container. In Figure 25.25 the direction of flow is shown by the blue arrows.

Simulation Contra

Controls for Fluid Simulation are found in the **Properties Editor, Physics Properties** (Figure 25.27). You have to set up both **Fluid Emitter Controls** and **Domain Controls.**

Figure 25.27

Fluid Emitter Controls

Deselect the Domain and select the **Fluid** Emitter Cube.

In the **Properties Editor, Physics Properties** click on **Fluid** (Figure 25.27) to display the **Fluid Tab**

In the **Fluid Tab** click where you see **Fluid Type: None** and select **Flow** in the menu that displays (Figure 25.28).

Fluid Settings display in the Editor.

In the **Fluid Tab** click where you see **Fluid Type: None** and select **Flow** in the menu that displays (Figure 25.28).

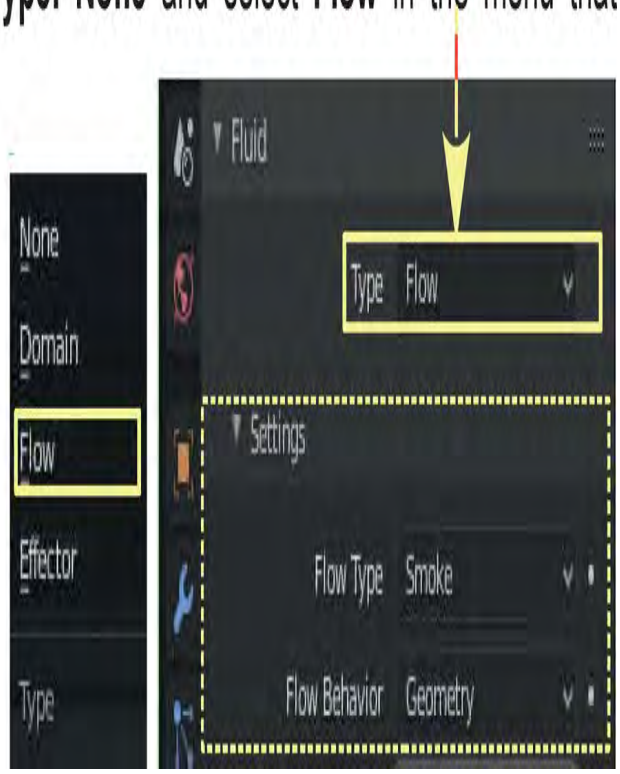


Figure 25.28

Two more settings to go: In the **Settings Tab** change **Flow Type:** Smoke to **Liquid** and **Flow Behaviour: Geometry** to **Inflow** (Figure 25.29).

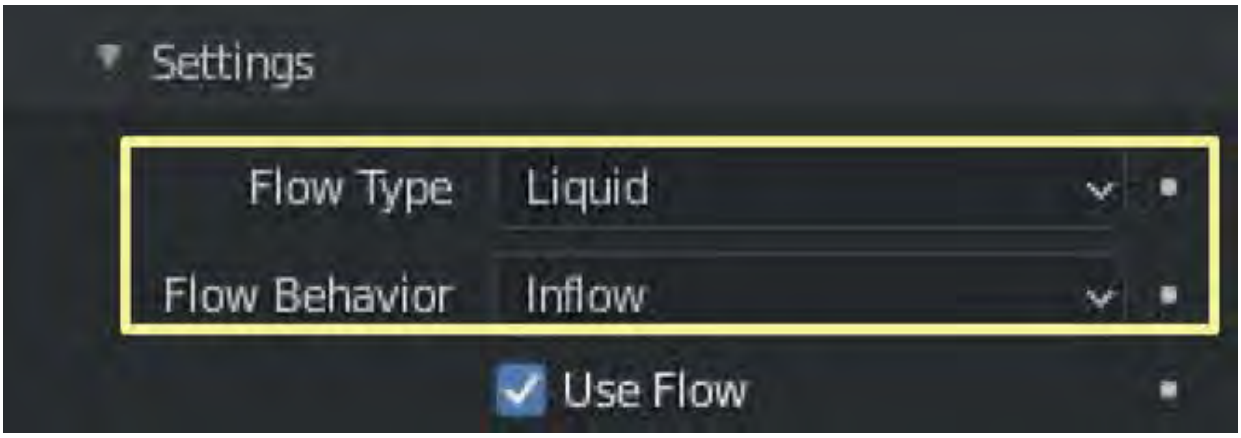


Figure 25.29

Check that **Use Flow** is checked (ticked).

Flow Behaviour

Flow Behaviour: Inflow sets the Emitter Object Cube to deliver a continuous flow of fluid. The flow of fluid is generated when an animation is played in the Timeline Editor. **Note:** At this point nothing will occur if the animation were to be played.

Figure 25.25 shows the fluid inflow direction, starting along the Y Axis of the 3D World. You set this by giving the fluid an initial velocity. In other words give the fluid a kick start in the Y direction. To do this check (tick) the **Initial Velocity Tab** and set the **Initial Y** value to **2 m/s.**

The completed set up for the Emitter Object in the Properties Editor is shown in Figure 25.30.

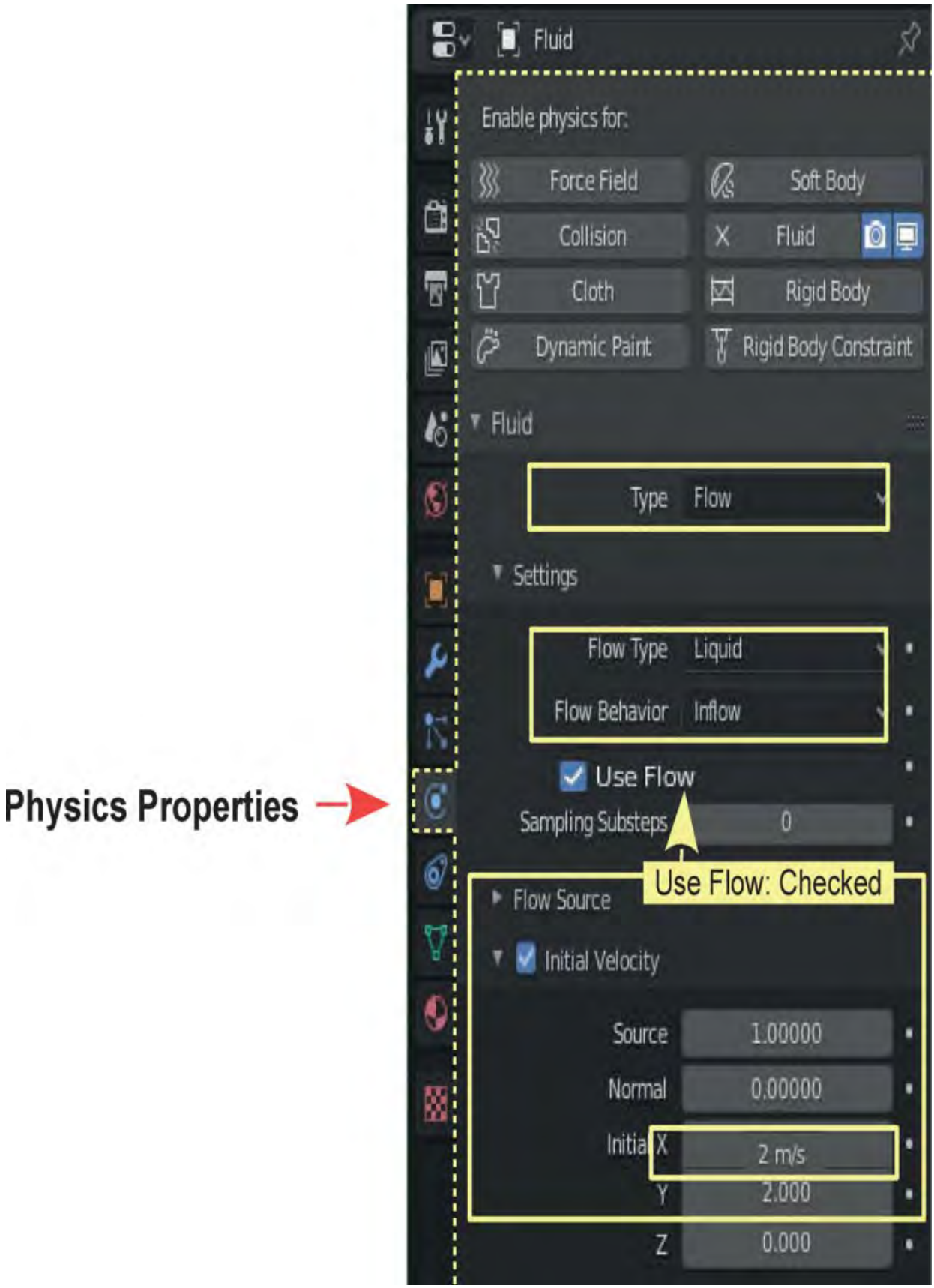

Physics Properties
Fluid
Enable physics for:
Force Field
Soft Body
Collision
X
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Fluid
Type
Flow
Settings
Flow Type
Liquid
Flow Behavior
Inflow
Use Flow
Sampling Substeps
0
Use Flow: Checked
Flow Source
Initial Velocity
Source
1.00000
Normal
0.00000
Initia
X
2 m/s
Y
2.000
Z
0.000

Figure 25.30

Domain Object Controls

The **Domain Control** set up is similar to the Emitter Control. With the Domain Cube selected, in the 3D Viewport Editor open the Physics Properties in the Properties Editor, click **Fluid.** This time change Fluid Type None to **Type: Domain.** In the **Settings Tab** change Domain Type from the default Gas to **Liquid** (Figure 25.31).

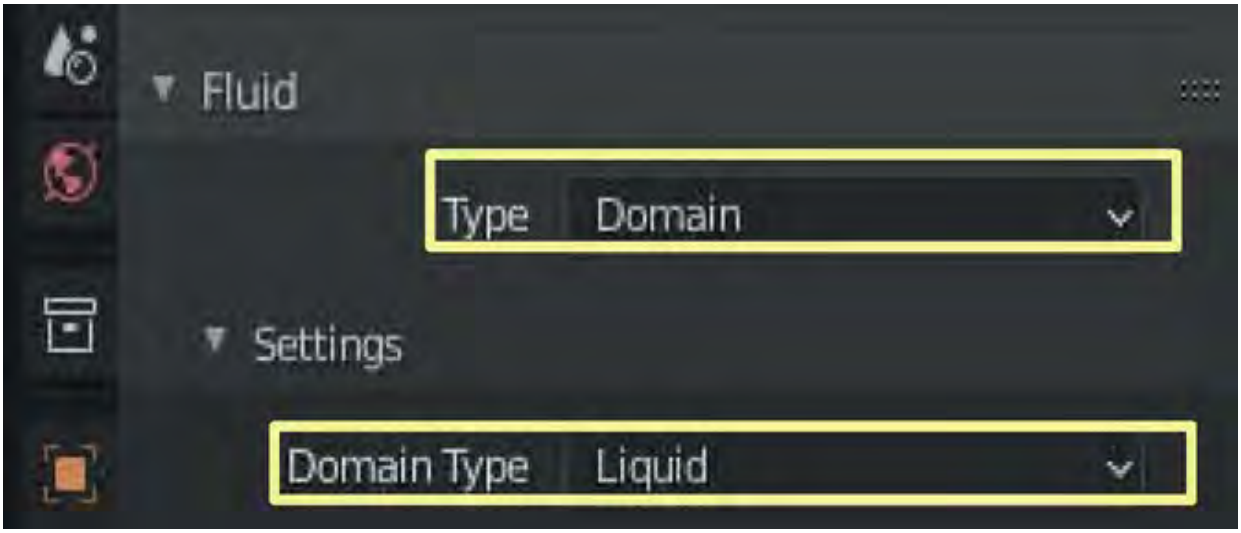


Figure 25.31

Obviously, setting Fluid Type: Flow with Flow Type: Liquid and Fluid Type: Domain with Flow Type: Liquid relates to the fact that this is a **Fluid Simulation** (Liquid) exercise. You will note that with the Fluid Emitter selected there are alternative Flow Type setting.

Playing the Animation

By playing the Animation in the **Timeline Editor** you will see Particles being Emitted from the Fluid Cube. The Particles initially move on the Y Axis until affected by Gravity when the fall and splash at the bottom of the Domain as if

it were a container. As the animation continues to play Particles are continually Emitted from the Fluid Cube. Particles splashing in the Domain appear to rise as they accumulate with the impression that they are the **upper surface of the liquid** in the Domain (Figure 25.32).

At this stage the Particles are displaying as a representation of the Fluid Flow only, not an actual display of Fluid. They merely give an indication of Fluid Flow.

Particles

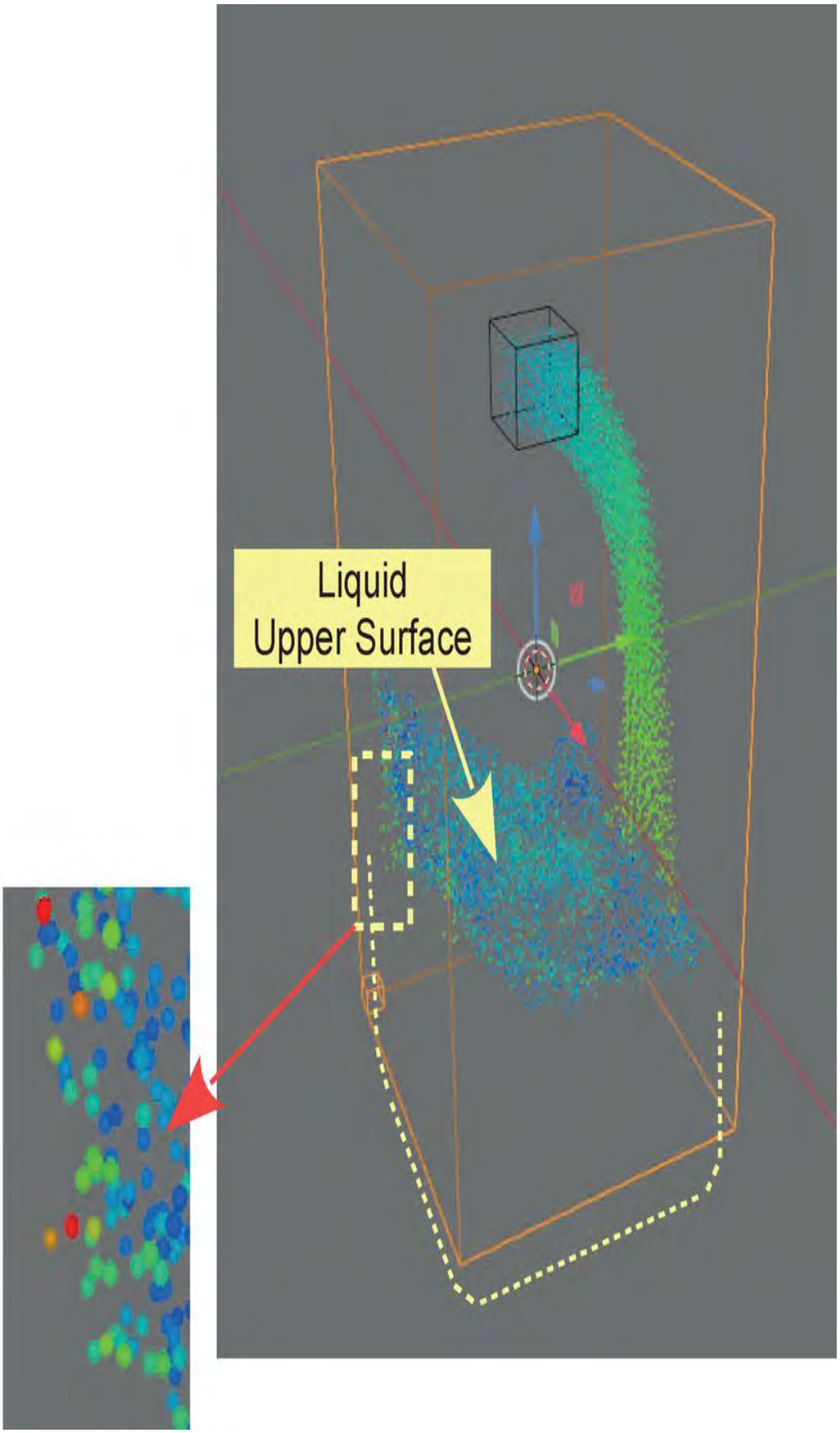

Liquid
Upper Surface
Particles

Figure 25.32

Particles

The Fluid Simulation relies on **Particles** (Chapter 24) being generated by a Particle System which obeys the Gravitational affect in the Scene and interact with Objects acting as obstacles or receptacles.

Remember: Particles Do Not Render.

At this point you may make adjustments to the control settings, reconfigure the Domain shape and resize and reposition the Fluid Flow Object to modify the Particle display. With each change you can replay the Animation to see the effect.

Memory Computer Power and Pakinn

As revisions and modifications are made you are working in the Blender program which, on opening, is placed in the computers memory (RAM). The Data generated producing the Particle display is, therefore, placed in the memory. This is not significant with the relatively simple Animation produced so far but when complex simulations are underway a significant amount of memory can be consumed which may hamper the computers ability to process information.

This being the case, in Fluid Simulation, the **Particle Display Data** is written to the hard drive in a **Cache Folder** to free memory for future processing.

The process of writing the data to the Hard Drive is termed **Baking.**

When you **Bake,** you should designate where on your hard drive to save the data. With the Domain selected in the 3D Viewport Editor, scroll way down to the bottom of the **Properties Editor, Physics Properties** to the **Cache Tab.** Blender automatically creates a folder for saving named **\cachefluid** which could have a File Path something like that shown in Figure 25.33 over.

Cache File Path

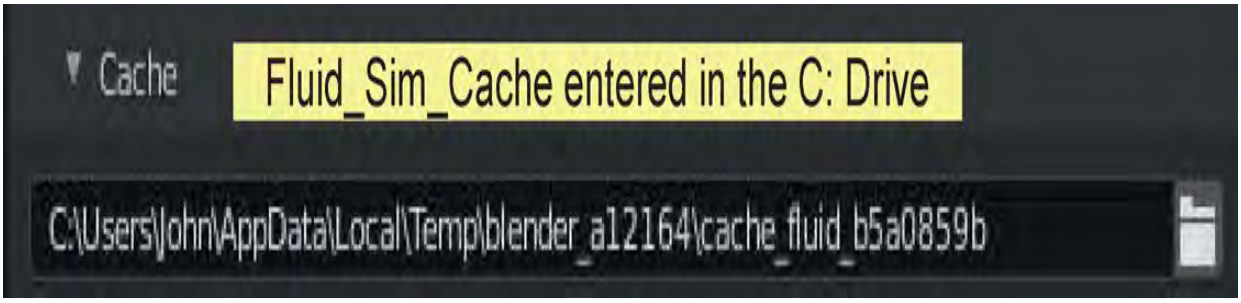


Figure 25.33

Obviously the **Cache File** (cache_fluid_b5a0859b) is not that easily accessed when searching for it on the computer. It is better to create a new Folder in a more accessible location.

Figure 25.34

To Bake the simulation to the Cache Folder you click the **Bake Button. Where is it?** At this stage the Bake Button or rather, Bake Buttons are hidden.

Cache Processing Controls

In the **Domain Properties, Cache Tab** you will see **Type: Replay.**

Repla is one of three Modes which determine if you can Bake or how you Bake. With **Replay** there are no Bake Buttons. **You cannot Bake.**

Modular: provides individual Bakes for Baking **Settings Data, Particle Data and Mesh Data.**

Bakes everything at the same Time.

When **Type: Replay** is active you may make adjustment to controls, reconfigure the arrangement and replay the Animation as previously mentioned but you **Can Not Bake.** To Bake you have to select **Type: Modular** or **Type All.**

Note: Mantaflow allows a Bake to be paused and restarted but only with **Is Resumable** checked. If **NOT** checked you see a warning.

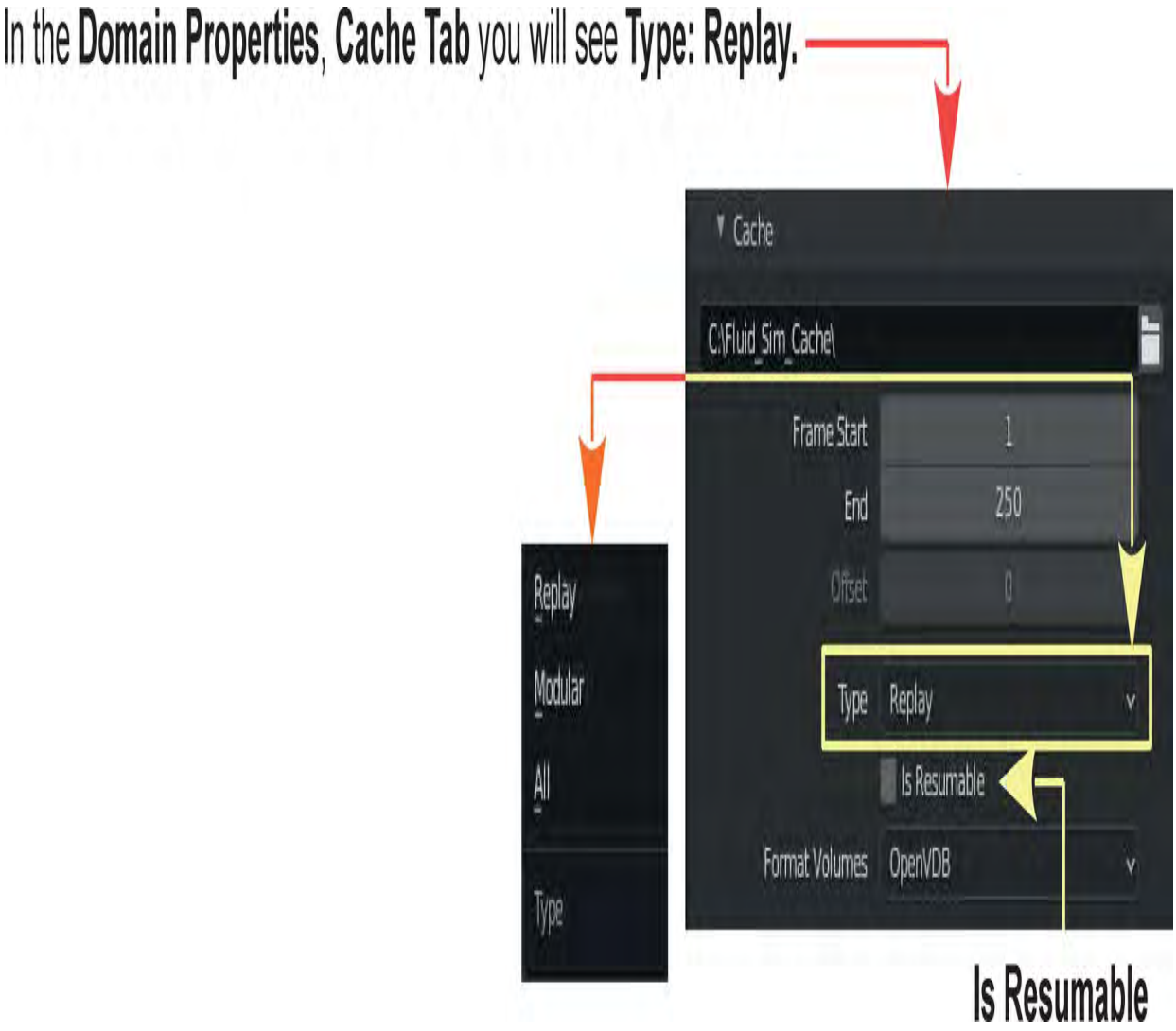


Figure 25.35

With **Is Resumable** checked you may pause the Bake by pressing **Esc.** In which case **Resume and Free** buttons display. Resume continues the Bake, Free clears the Cache.

Figure 25.36

With the Bake paused you change back to Replay Mode to rerun the Animation when adjustments have been made.

Click **Resume** to continue Baking

Fine Tuning the Flow

While **Cache Type: Replay** remains active run the Animation to check the Fluid Flow. Before adjusting settings to modify or fine tune the Flow (with Cache Type: Replay and Is Resumable checked) make sure the designated Cache Folder is empty.

If you are setting up a new Simulation having previously created one and have the old Cache Folder selected in the Cache Tab, Blender will attempt to use the files in the Cache Folder. This may result in the new Simulation not playing or you could experience unexpected results.

If you have been experimenting you will inevitably generate multiple folders and at times Blender may attempt to use data previously stored.

Setting Examples

Time Scale: Determines the speed of Flow (Figure 25.38).

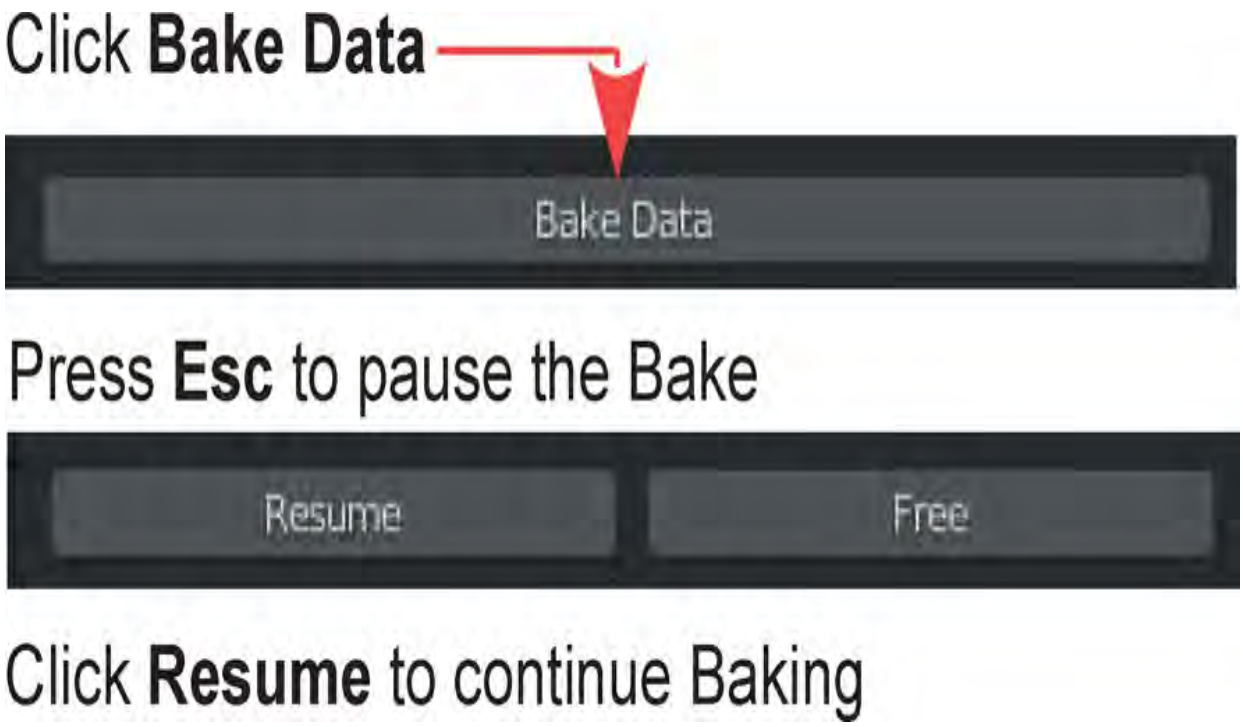


Figure 25.37

Border Collisions: Check or uncheck to set how Particles react with the Domain boundaries (Figure 25.39).

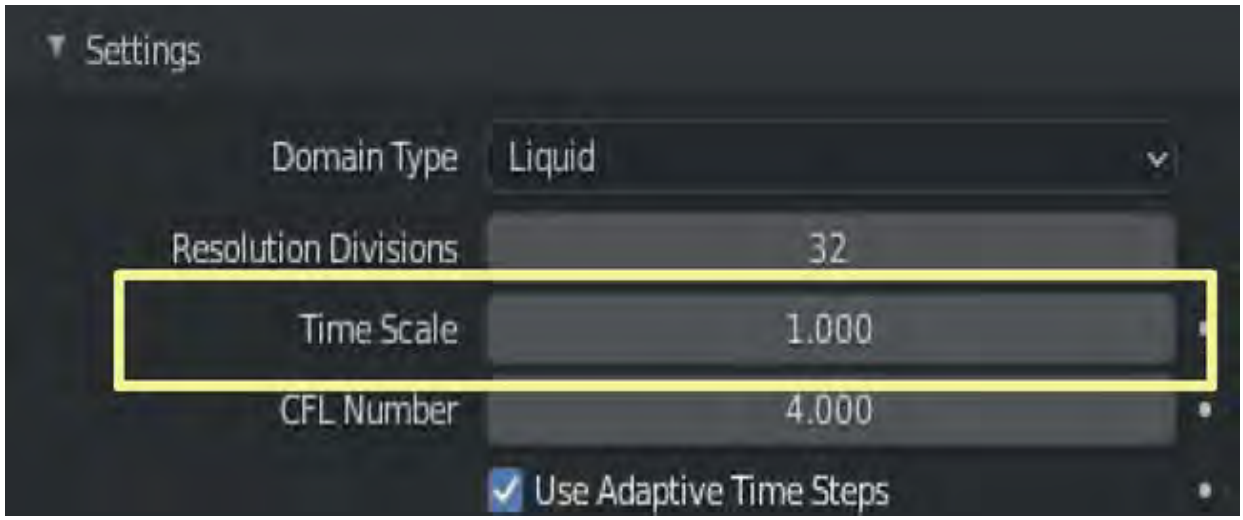


Figure 25.38

Liquid – Particle Maximum_ regulates the number of Particles produced (Figure 25.40)

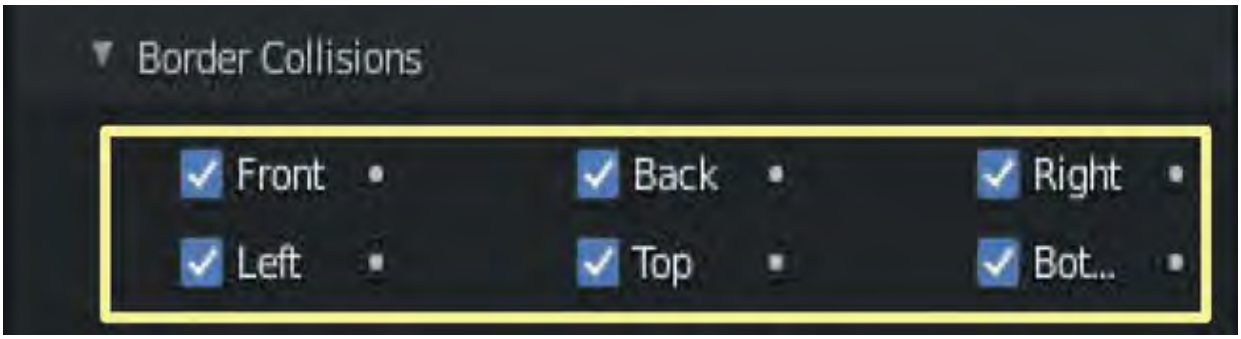


Figure 25.39

This is a sample of settings only to make you aware of the controls available for regulating the Fluid Flow.

When satisfied with the Fluid Flow it is time to Bake the Data which saves the information about the Particles to the Cache. Remember; the Particles do not Render, therefore, there is a way to go before a final Fluid Simulation is achieved. Baking can take a considerable time, depending on the complexity of the Fluid Simulation. At this point in the demonstration there is only an Emitter and a Domain. There are additional control Objects which may be introduced. This will be covered later.

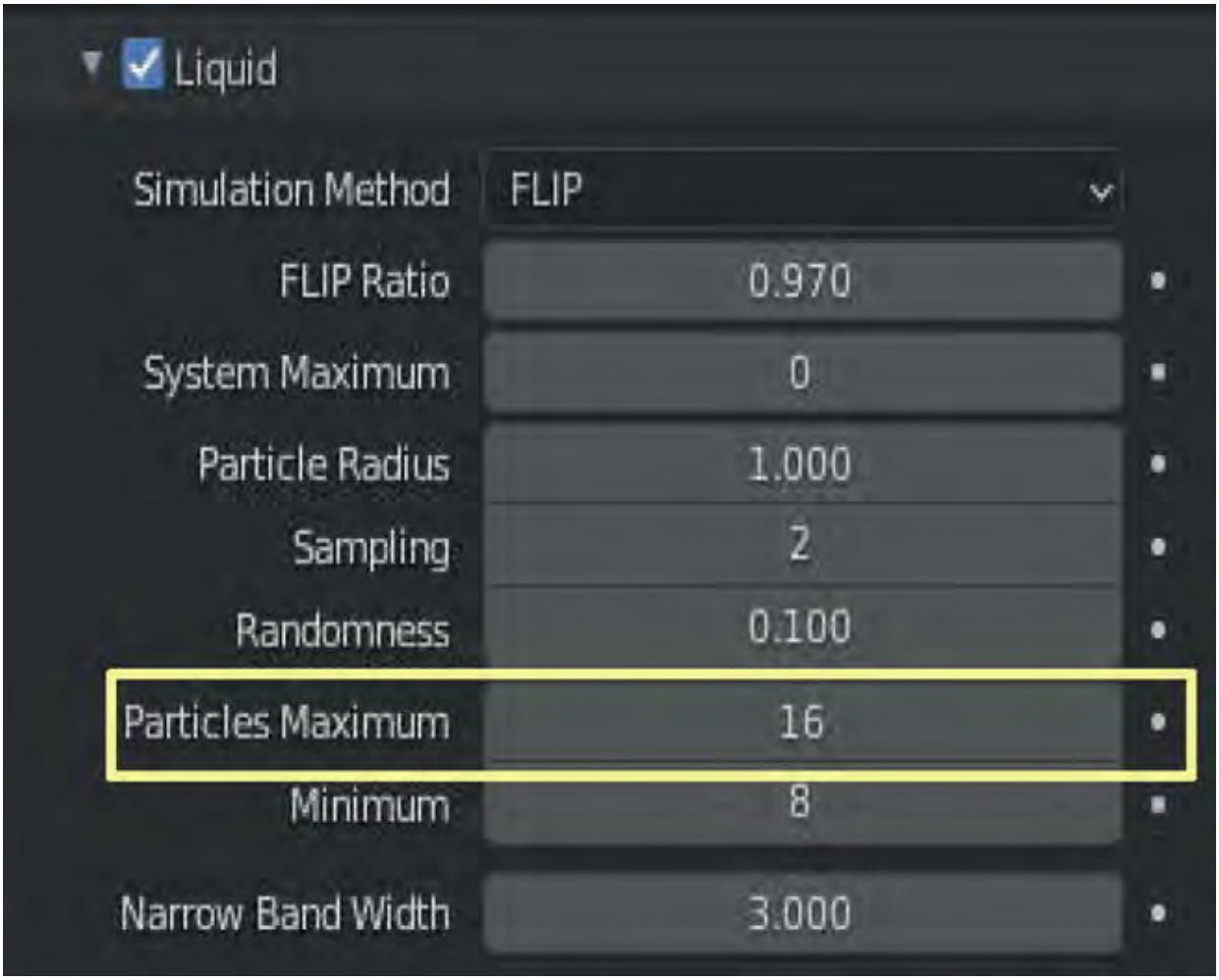


Figure 25.40

At the moment the Fluid Flow comprises Particles for Fluid only. Other Particles may be included for Spray, Foam and Bubbles (Figure 25.41).

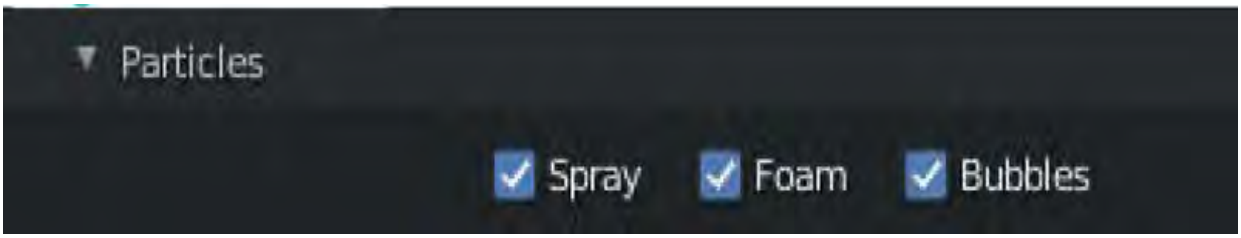


Figure 25.41

Limiting Bake Time

For experimenting you may limit the Bake Time. 50 Frames will keep the Bake time reasonable.

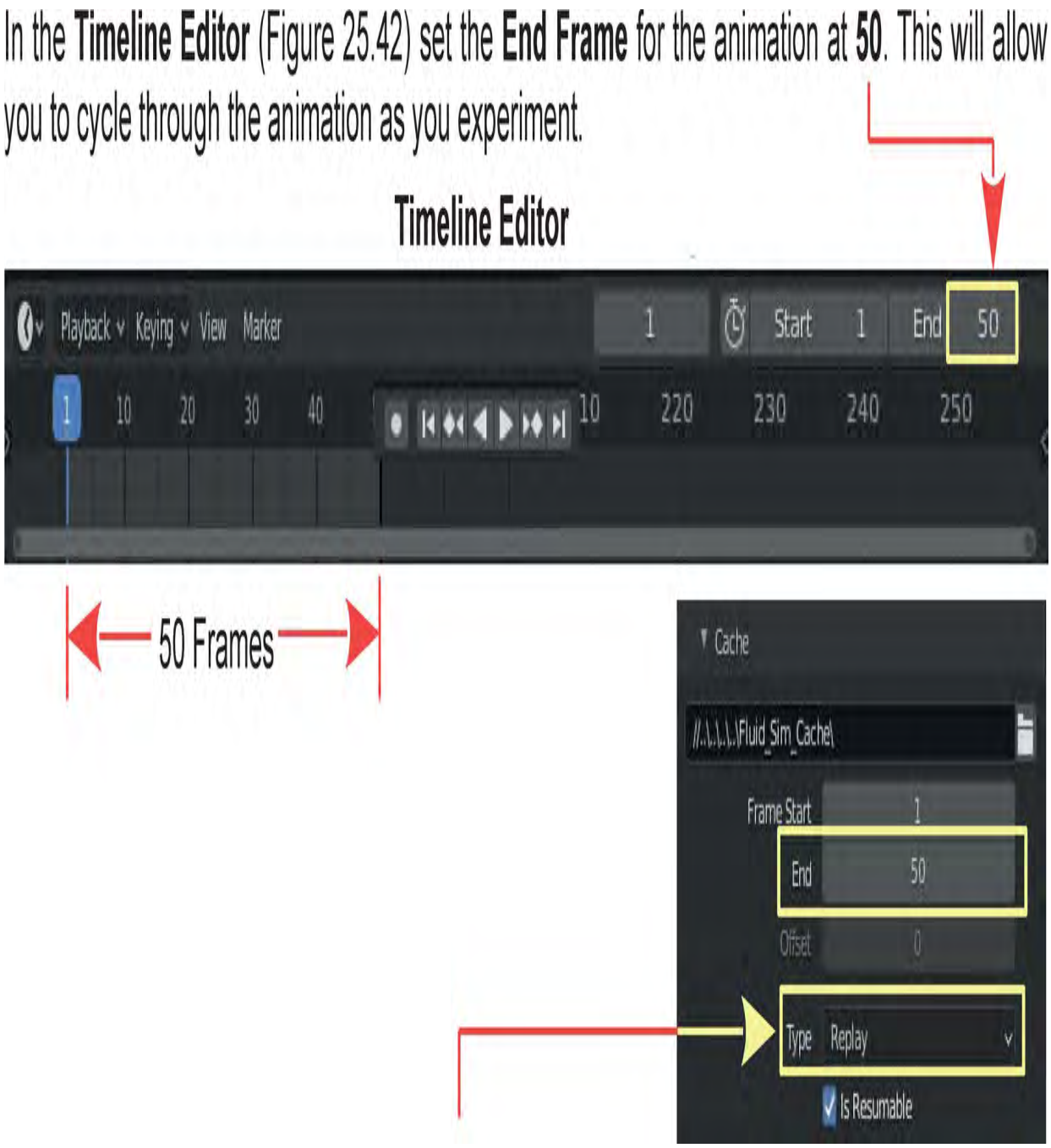


Figure 25.42

In the Properties Editor, with the Domain selected go to the Cache Tab and set the **Frame End** to 50. With **Type: Replay** in the Cache Tab check the Fluid Flow in the 3D Viewport Editor (Play the Animation).

Change Type : Replay to Modular.

Withe the Domain selected in the 3D Viewport Editor click **Bake Data** in the Physics Properties, Settings Tab. The Bake commences with a **Progress Bar** displayed at the bottom of the Screen.

Figure 25.43

When the Bake is complete you the Emitter Cube (Figure 25.44). see Particles consolidated on Play the animation in the Timeline Editor to see the Particles emitted and cascade into the Domain (Figure 25.45).

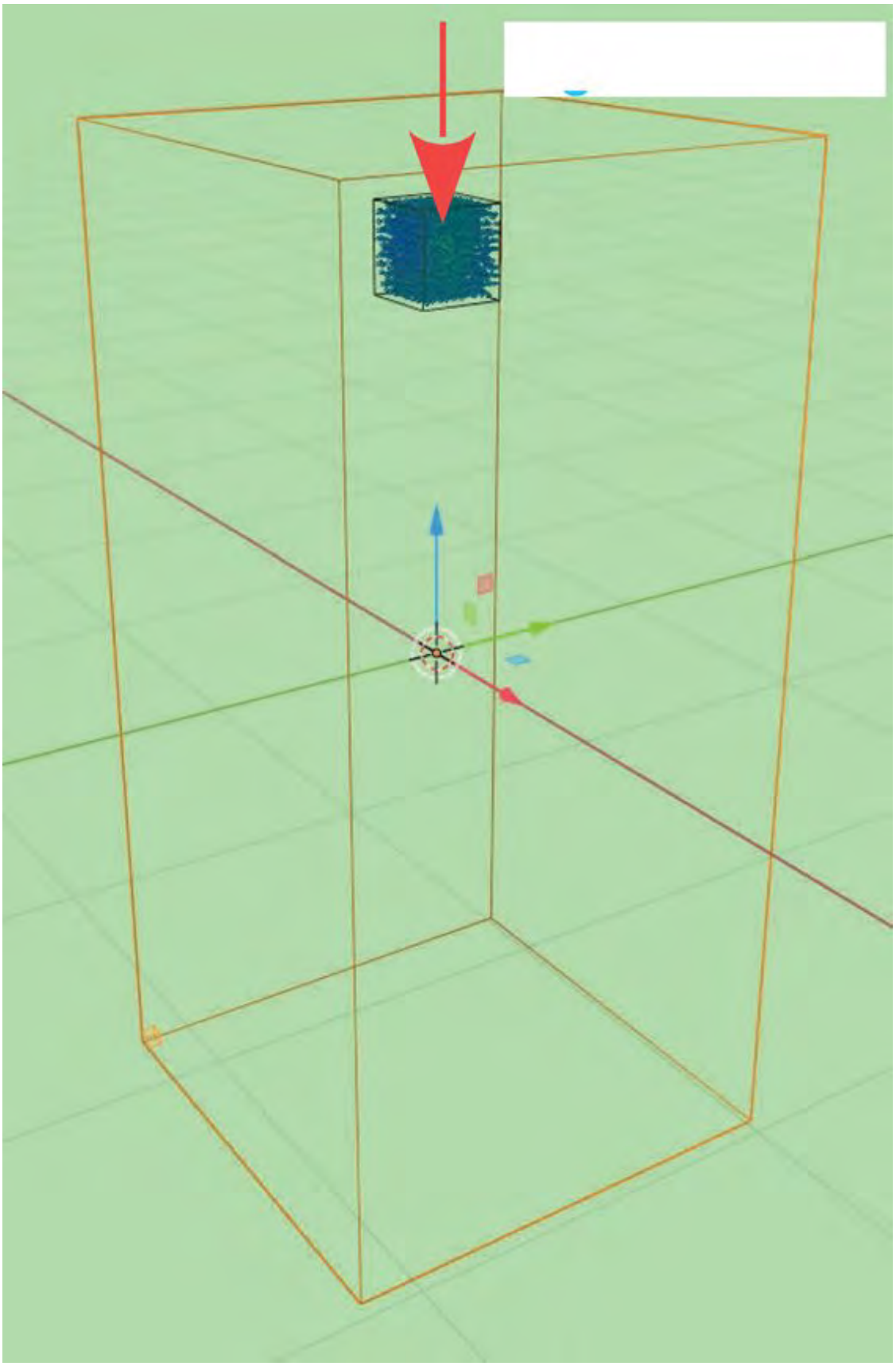
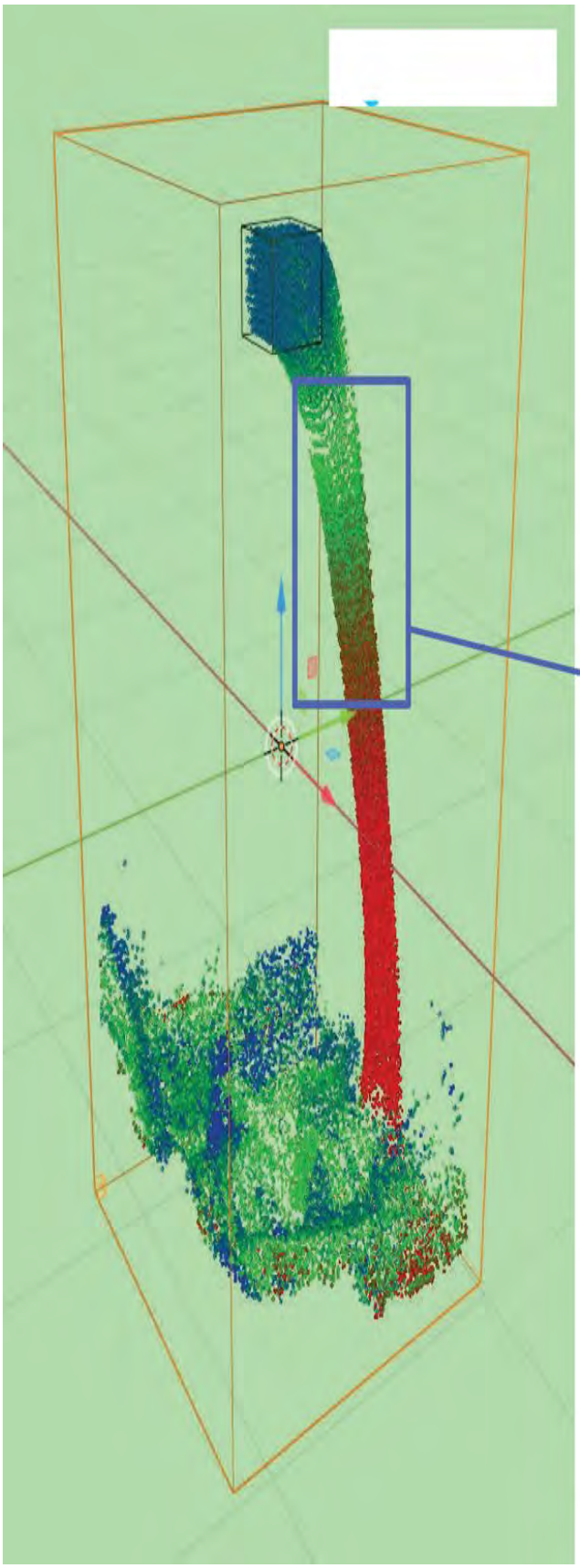
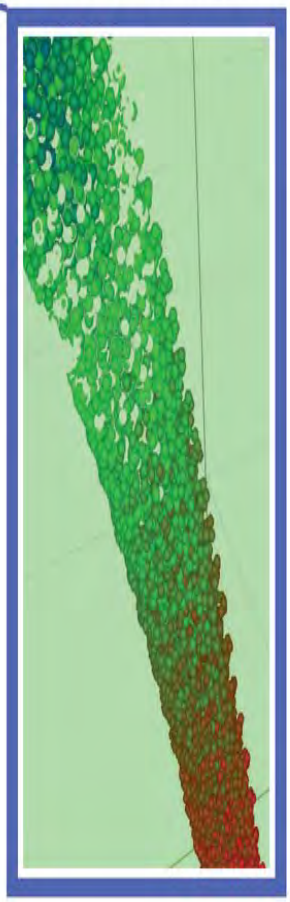


Figure 25.44



Remember: Particles do NOT Render.



Rendered View

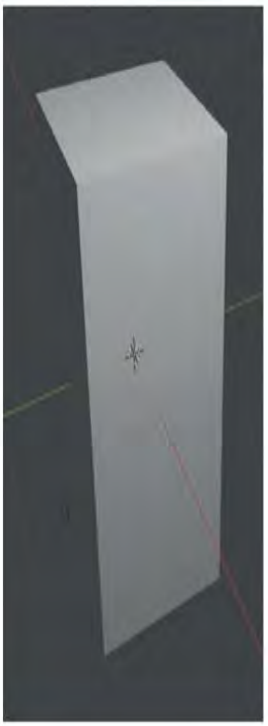


Figure 25.45

Bake Mesh

Having Baked the Data creating Particles that define the Fluid Flow in an animation you **Bake Data to create a Mesh.** The Particles will not Render but the Mesh does and the Mesh shape will be animated.

To Bake the Mesh have the Domain selected in the 3D Viewport Editor and in the Properties Editor, Physics buttons **check Mesh** in the Mesh Tab (Figure 25.46).

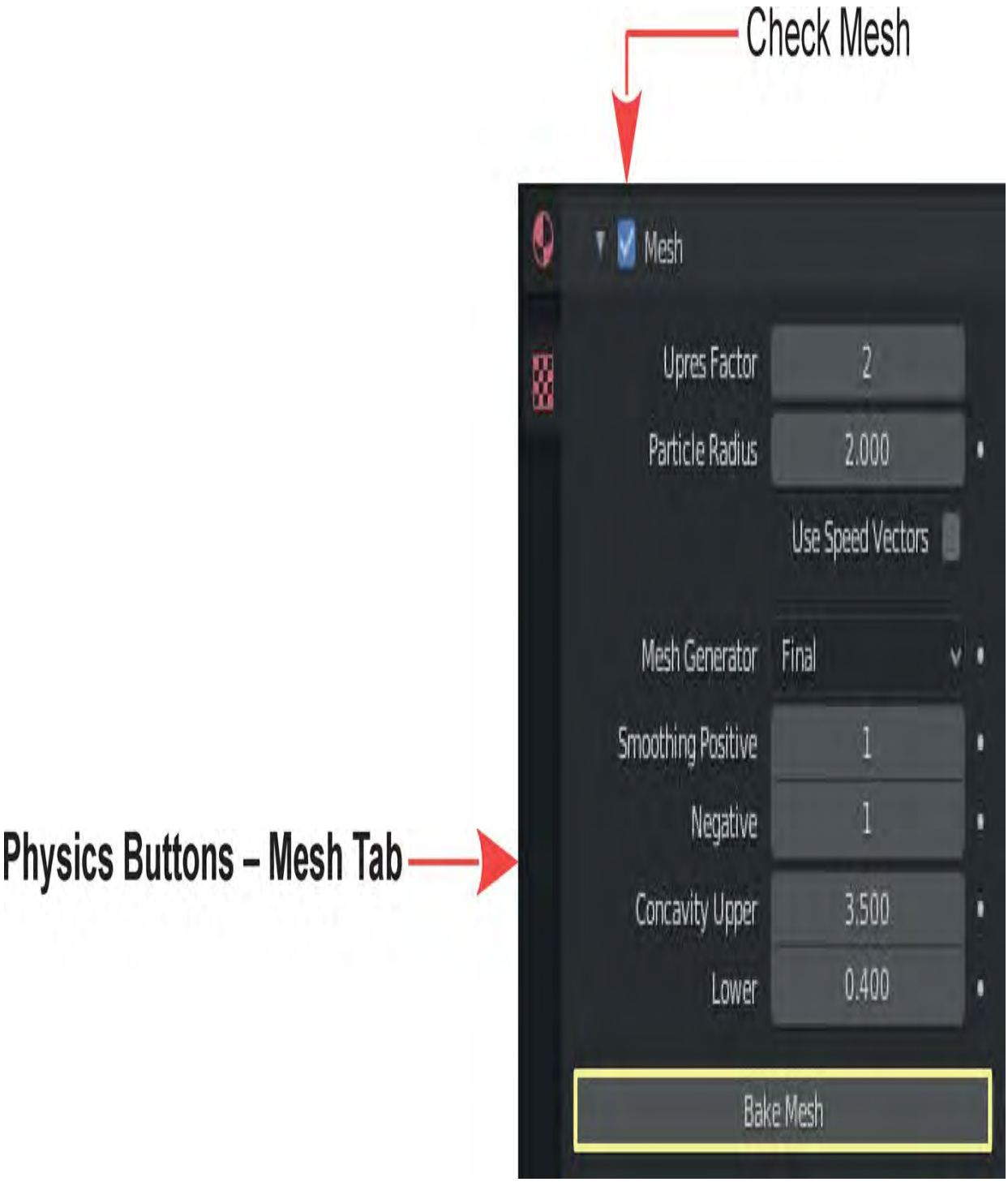
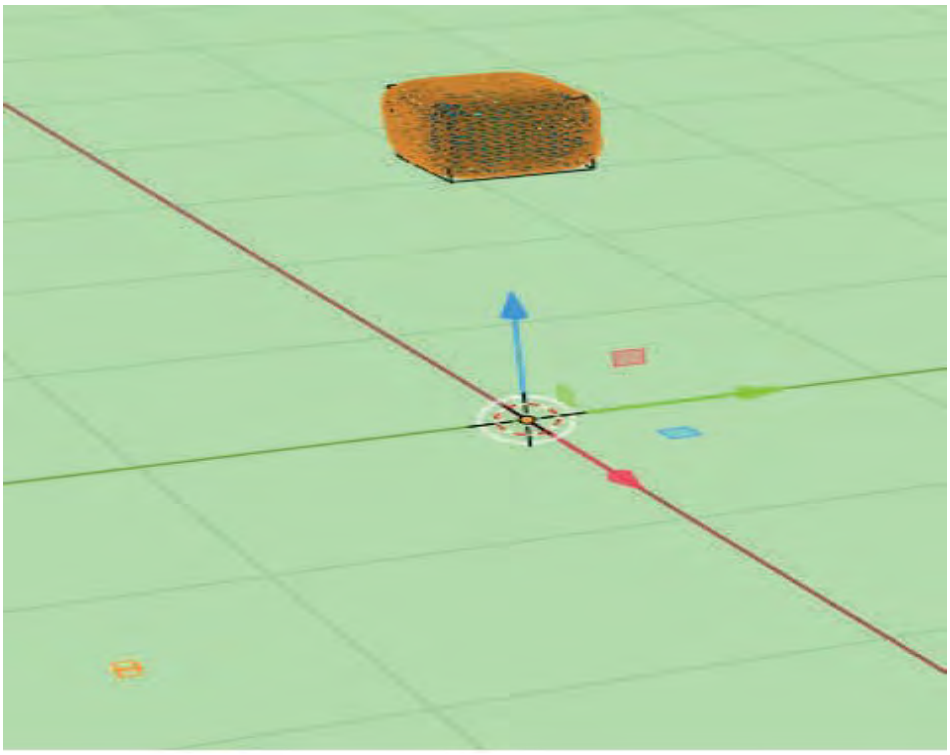

Check Mesh
Mesh
Upres Factor
2
Particle Radius
2.000
Use Speed Vectors
Mesh Generator
Final
Smoothing Positive
1
Negative
1
Concavity Upper
3.500
Lower
0.400
Bake Mesh
Physics Buttons – Mesh Tab

Figure 25.46



Rendered View

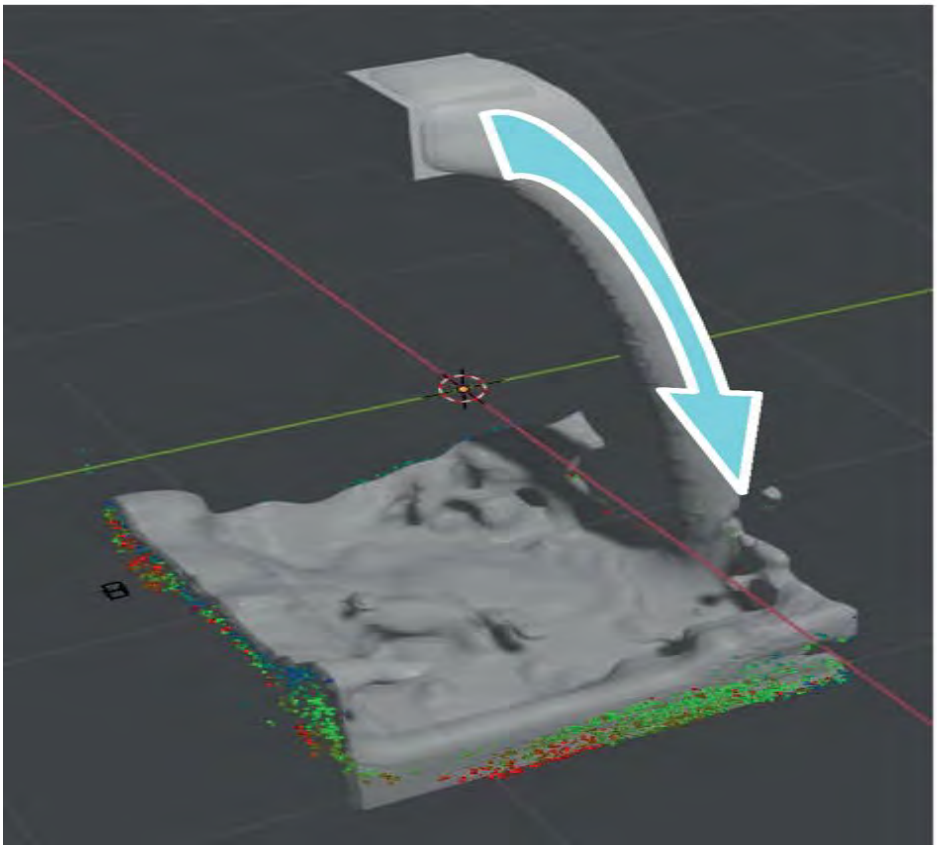
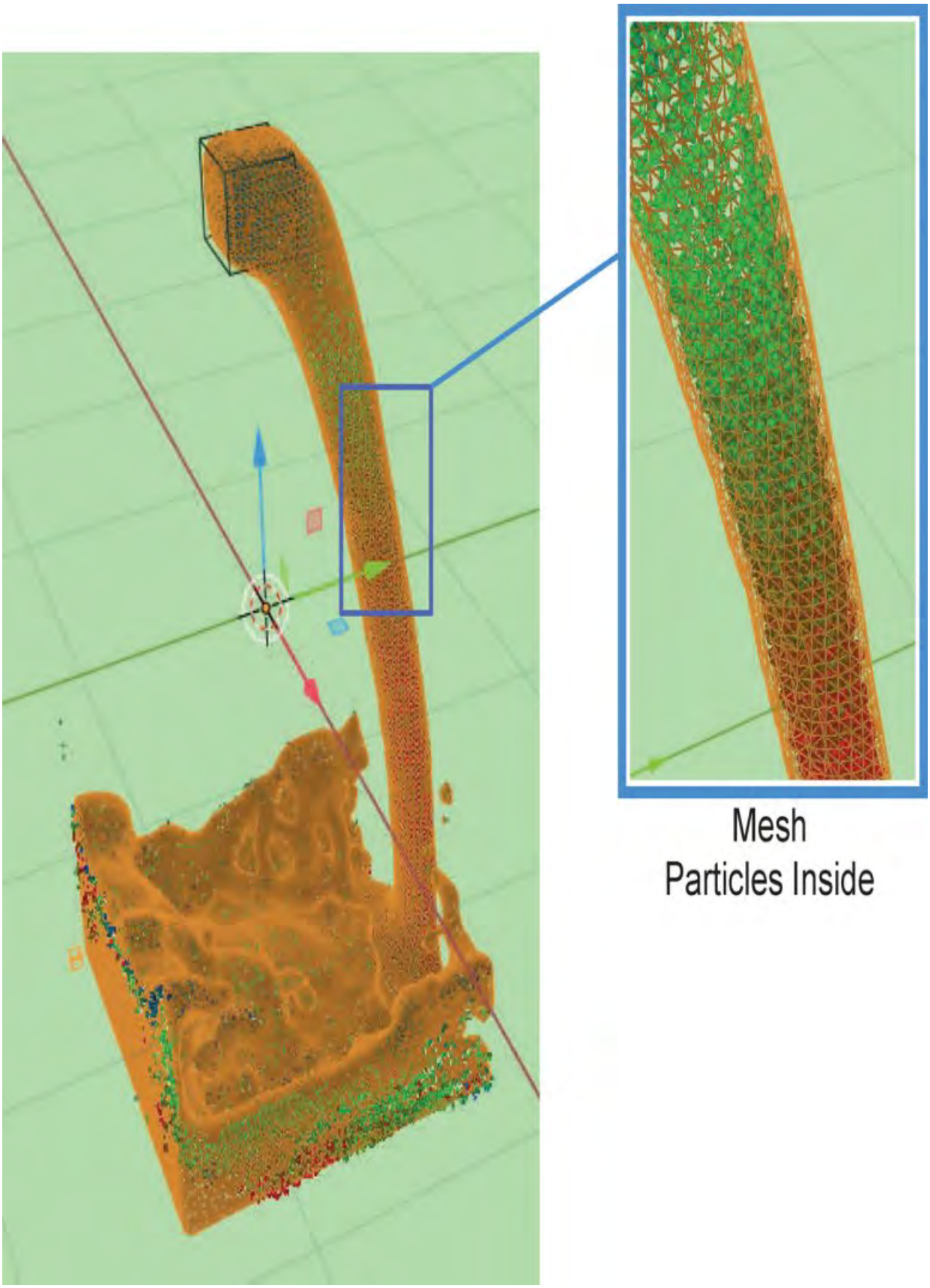


Figure 25.47



Mesh
Particles Inside

Figure 25.48

Click **Bake Mesh** to start the Bake and display the progress bar at the bottom of the Screen.

When the Bake is complete the Mesh displays in the 3D Viewport Editor consolidated on the Emitter Cube (Figure 25.47).

Replay the animation to see Particles surrounded by a Mesh (Figure 25.48).

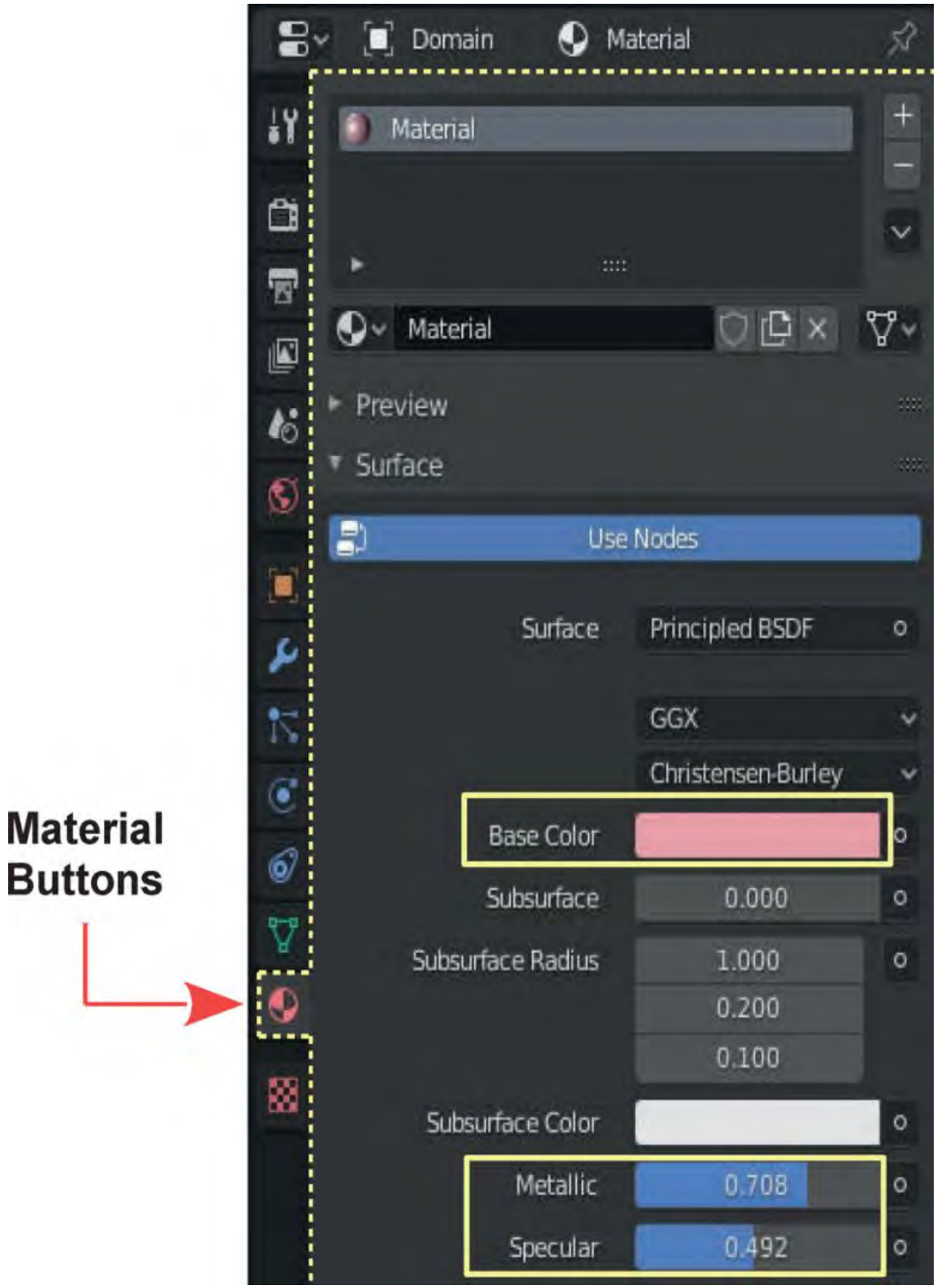

Material Buttons
Domain
Material
Material
Material
Preview
Surface
Use Nodes
Surface
Principled BSDF
GGX
Christensen-Burley
Base Color
Subsurface
0.000
Subsurface Radius
1.000
0.200
0.100
Subsurface Color
Metallic
0.708
Specular
0.492

Figure 25.49

Play the animation with the 3D Viewport Editor in **Rendered Viewport Shading Mode** to see the Fluid Flow.

At this point the Rendered View shows the Fluid Mesh as a gray color. Since the demonstration began by using the default Cube Object as the Domain the default gray Material is displayed. With a few tweaks in the Material buttons (Figure 25.49) you can have hot chocolate (Figure 25.50).

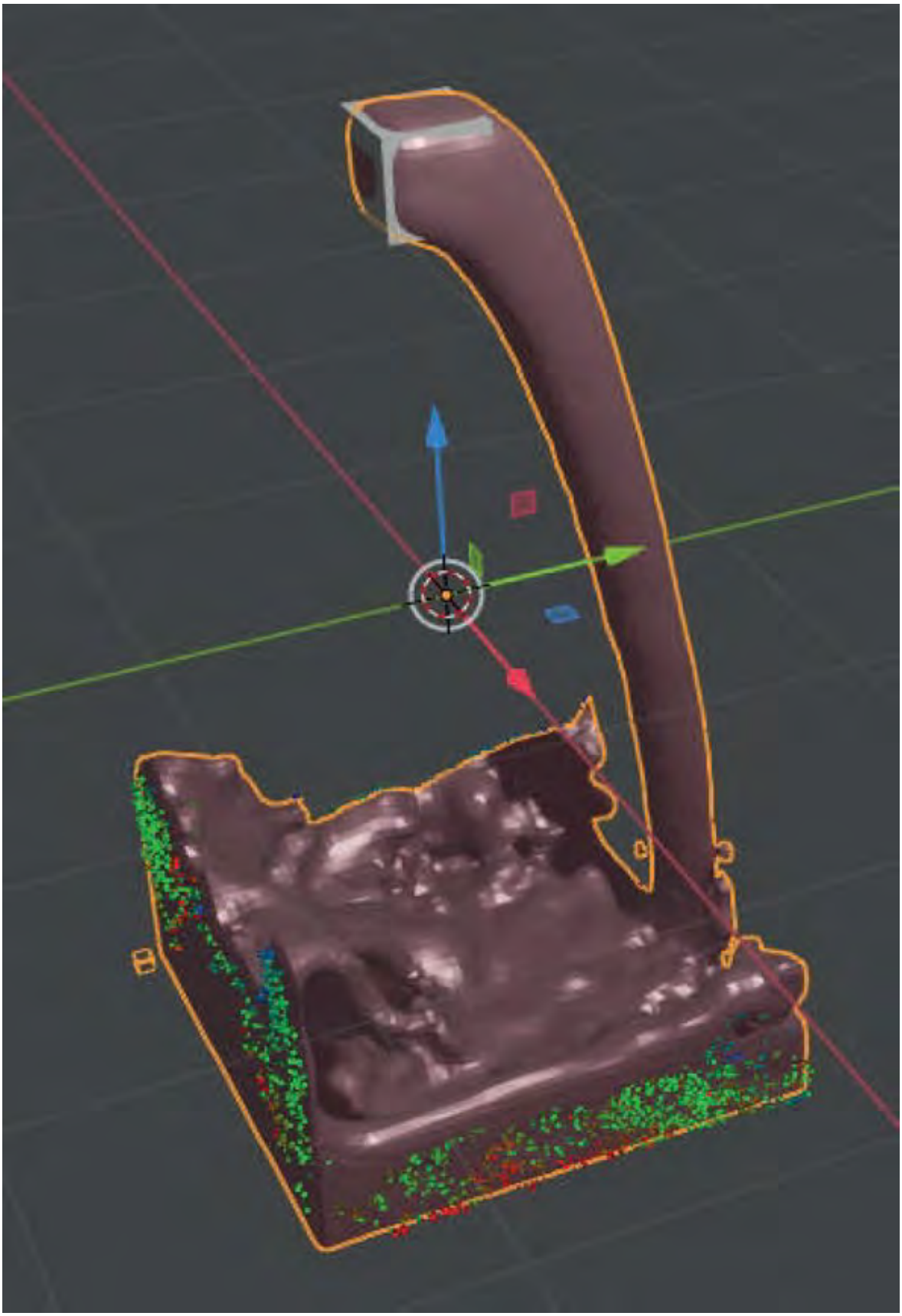


Figure 25.50

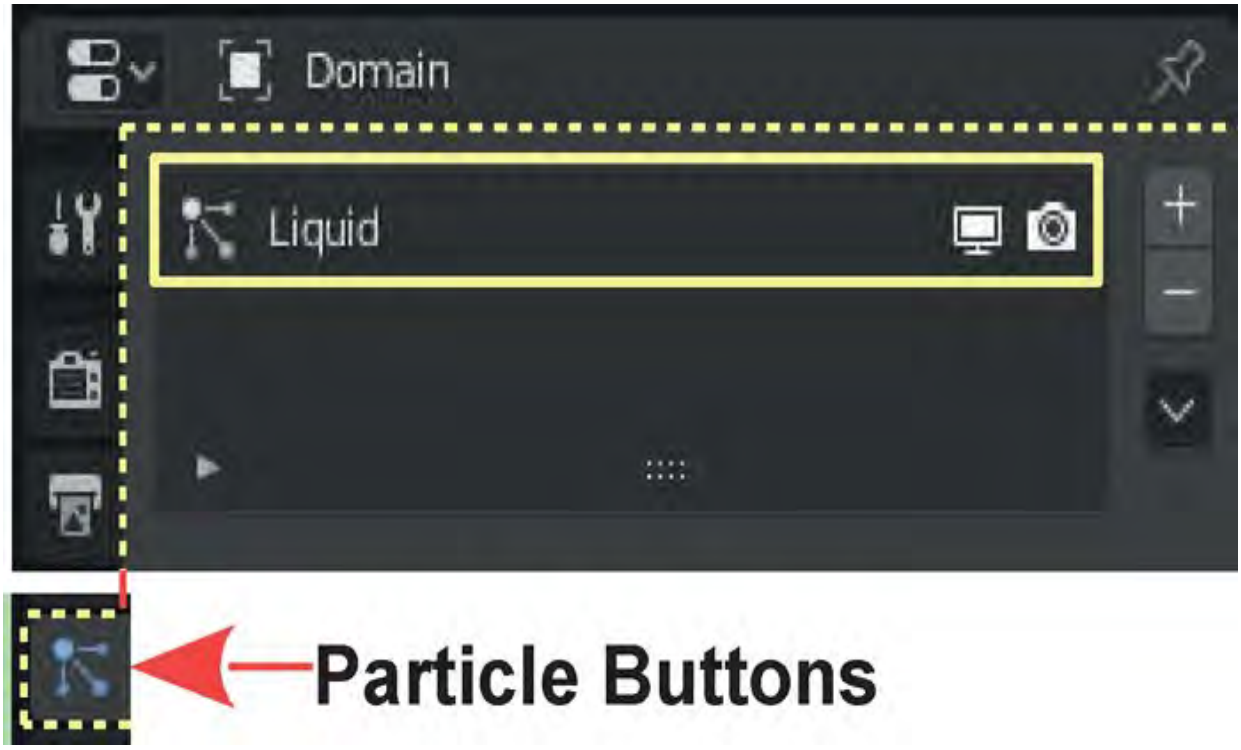


Figure 25.51

Remember: If your Domain is not created from the default Cube Object you will have to apply a New Material.

Rebaking

You may pause a Bake at any time by pressing the **Esc Key.** Pausing allows you to preview the Fluid being generated in the 3D Viewport Editor. If it looks OK you can continue the Bake. The Bake buttons in the Properties Editor will display a **Resume** button to allow you to restart the Bake. **Note:** As soon as you start a Bake the Physics button Controls become grayed out. At this point you can not change settings. The Bake Data and Bake Mesh buttons change to **Free Data** and **Free Mesh.** Clicking either of these cancels the data saved to memory in the Cache and makes the controls active. You may

then adjust settings and Rebake. You may Rebake the Mesh without Rebaking the Data.

As previously stated, **Baking** writes Data to the Cache Folder on the hard drive to save RAM. When you save the Blender File the Data written to the Cache Folder is not saved with the Blender File. When you close the Blender File the Data remains in the Cache on the hard drive. If you create a new simulation using the same Cache Folder the Data will be overwritten, therefore, the simulation in the original Blender file will not play when the file is reopened. **Therefore, create a new Cache Folder for a new simulation.**

Particles on Particles

So far Particles have been generated which simulate a fluid flow. These were created when the Data was Baked. In the Properties Editor, Particle buttons, with the Domain selected. You will see that a **Particle System named Liquid** has been automatically applied to the Domain.

Note: You can not edit this Particle System in the Particle buttons.

Figure 25.52

Additional Particles may be created to act as **Spray, Foam** or **Bubbles.** To create these Particles you check either one or all of the buttons in the **Properties Editor, Fluid Physics buttons, Particles Tab** (Figure 25.52). The controls in the Particles may be adjusted to affect the selection as a whole.

Having checked a selection you will find that a Particle System has been automatically created and added to the Properties Editor, Particle Buttons tab.

In Figure 25.52 **Particle Type: Bubbles** is selected in the Particles Tab.

In Figure 25.53 you see the **Particle System** named **Bubbles** has been automatically created.

Remember: Particles do not Render.

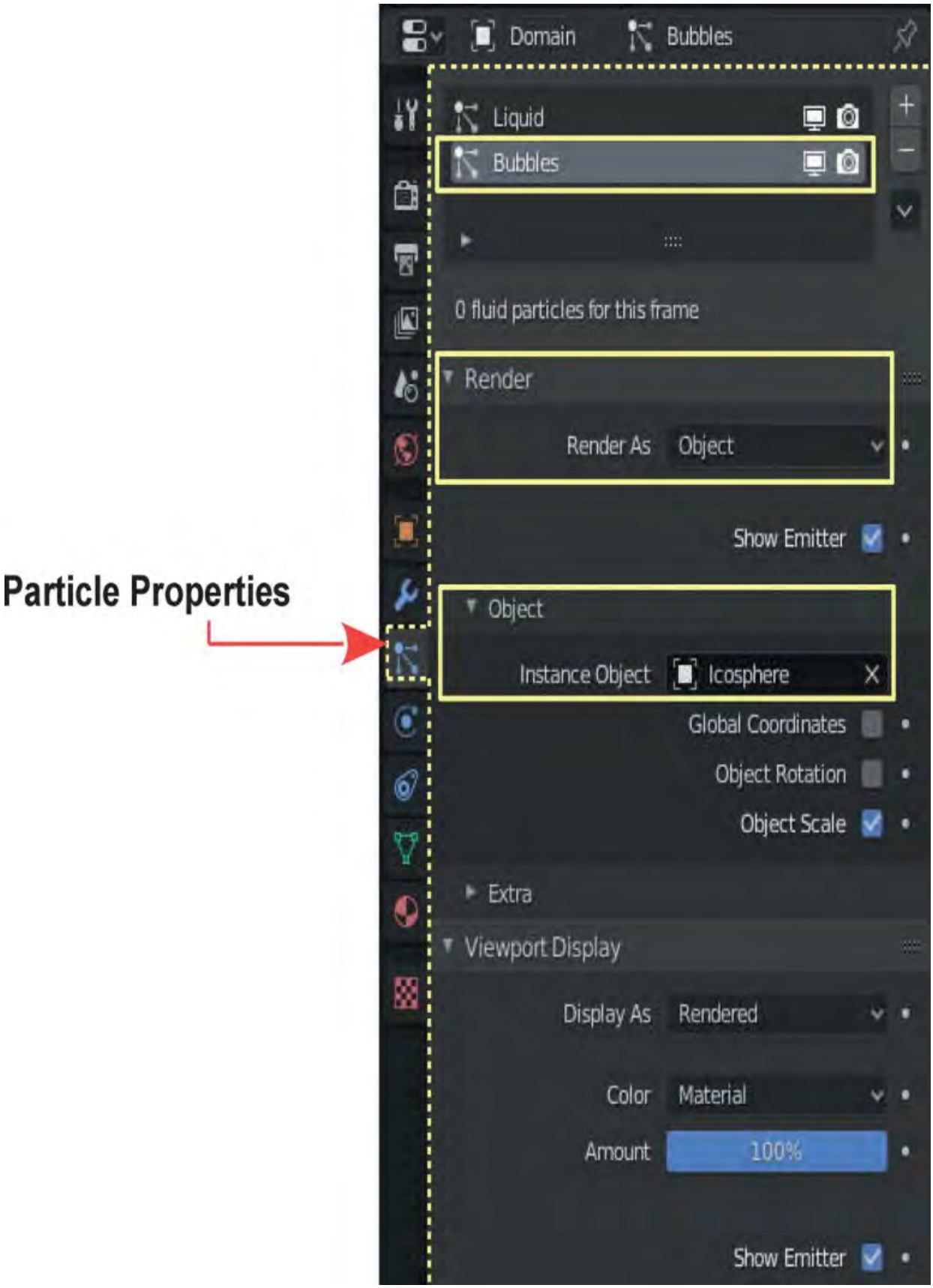

Particle Properties
Domain
Bubbles
Liquid
Bubbles
0 fluid particles for this frame
Render
Render As
Object
Show Emitter
Object
Instance Object
Icosphere
Global Coordinates
Object Rotation
Object Scale
Extra
Viewport Display
Display As
Rendered
Color
Material
Amount
100%
Show Emitter

Figure 25.53

To have the Bubbles display amongst the fluid in the 3D Viewport Editor you add an Object to the Scene and set the Particle System to Render the Particles (display) as that Object.

In this exercise the Object will be an **Icosphere.**

Summary

In following this exercise, at this juncture, you may be in danger of loosing your way, therefore, a summary follows.

The default Cube Object has been assigned as the Domain.

A second Cube has been scaled down relative to the Domain and positioned inside the Domain. This Cube has been designated as the Inflow Object.

In the Properties Editor, Physics buttons, with the Domain selected, Fluid Data and Mesh Data have been Baked.

In Baking the Data the length of the animation has been set as 50 Frames, therefore, in the Cache Folder you will find three sub folders named; **config** and **data** for the Data Bake and **mesh** for the Mesh Bake. These folders contain 50 x **.uni** files in the case of the Data Bakes and 50 x **.bobj** files for the Mesh Bake.

To continue; Add an **Icosphere** to the Scene in the 3D Viewport Editor. When you add the Icosphere set the **Subdivisions** in the **Last Operator panel** to 1, the objective being to have an Object with a minimal number of vertices which the Particles will render as. This minimises the number of calculations for the computer when Baking Data.

Add a nice bright Material to the Icosphere and set Shading to Smooth. Park the Icosphere off to one side in the Scene (Figure 25.54).

In the **Properties Editor, Particles Properties** (with the Domain selected) (to select the Domain click on Domain in the Outliner Editor) select (click on) the **Particle System named Bubbles** (Figure 25.41). In the **Render Tab** set **Render As** to **Object** and in the **Object Tab** set **Instance Object** to **Icosphere** (Figure 25.53).

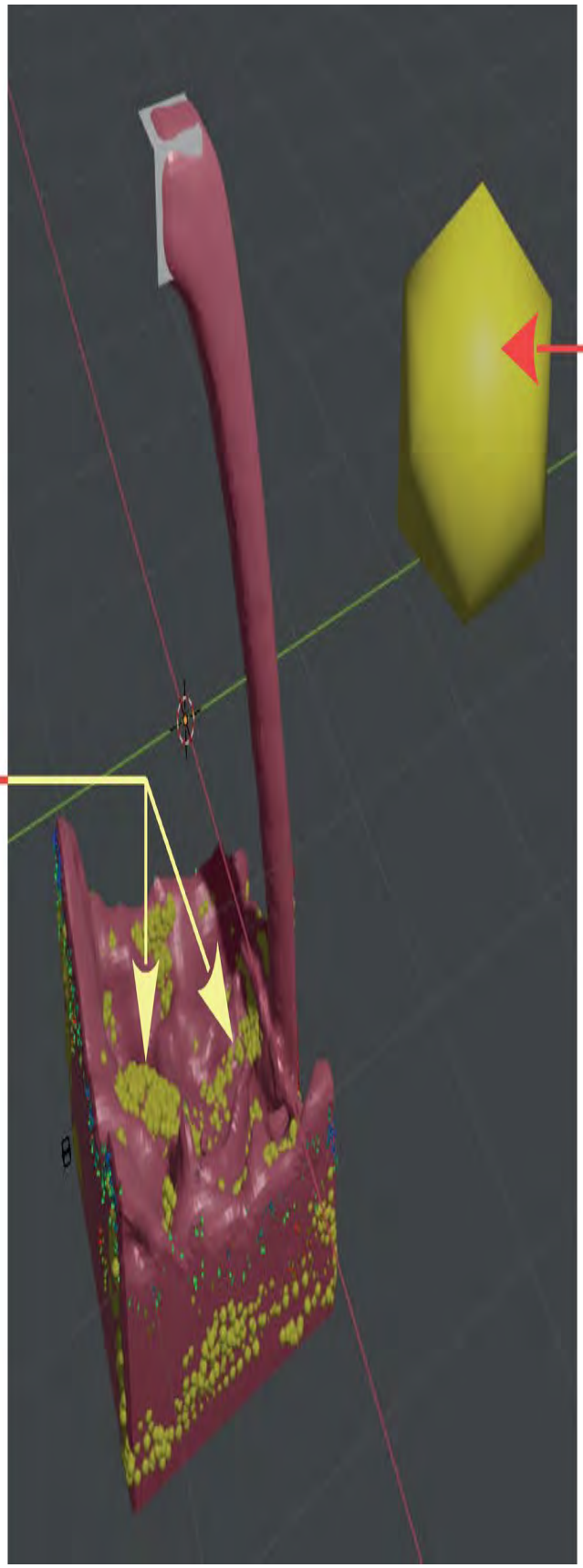
In the **Viewport Display Tab** set **Display As** to **Rendered.**

Remind yourself that this is fun and when you become conversant with the settings you will produce fantastic results.

Bake your Bubbles. In the Properties Physics buttons, Particles tab click **Bake Particles.**

You could have **Spray Particles** and **Foam Particles** set up and Bake these at the same time.

With the Bake complete, play the animation and pause at a Frame (Figure 25.54).



Icosphere parked to one side

Bubble Particles displaying as the Icosphere

Figure 25.54

In this exercise settings have been kept to a minimum. Obviously there are many many settings with which to experiment, but the foregoing should help in understanding the basic procedure for Fluid Simulation.

Before you go! Look at **Effector – Collision and Outflow Objects**

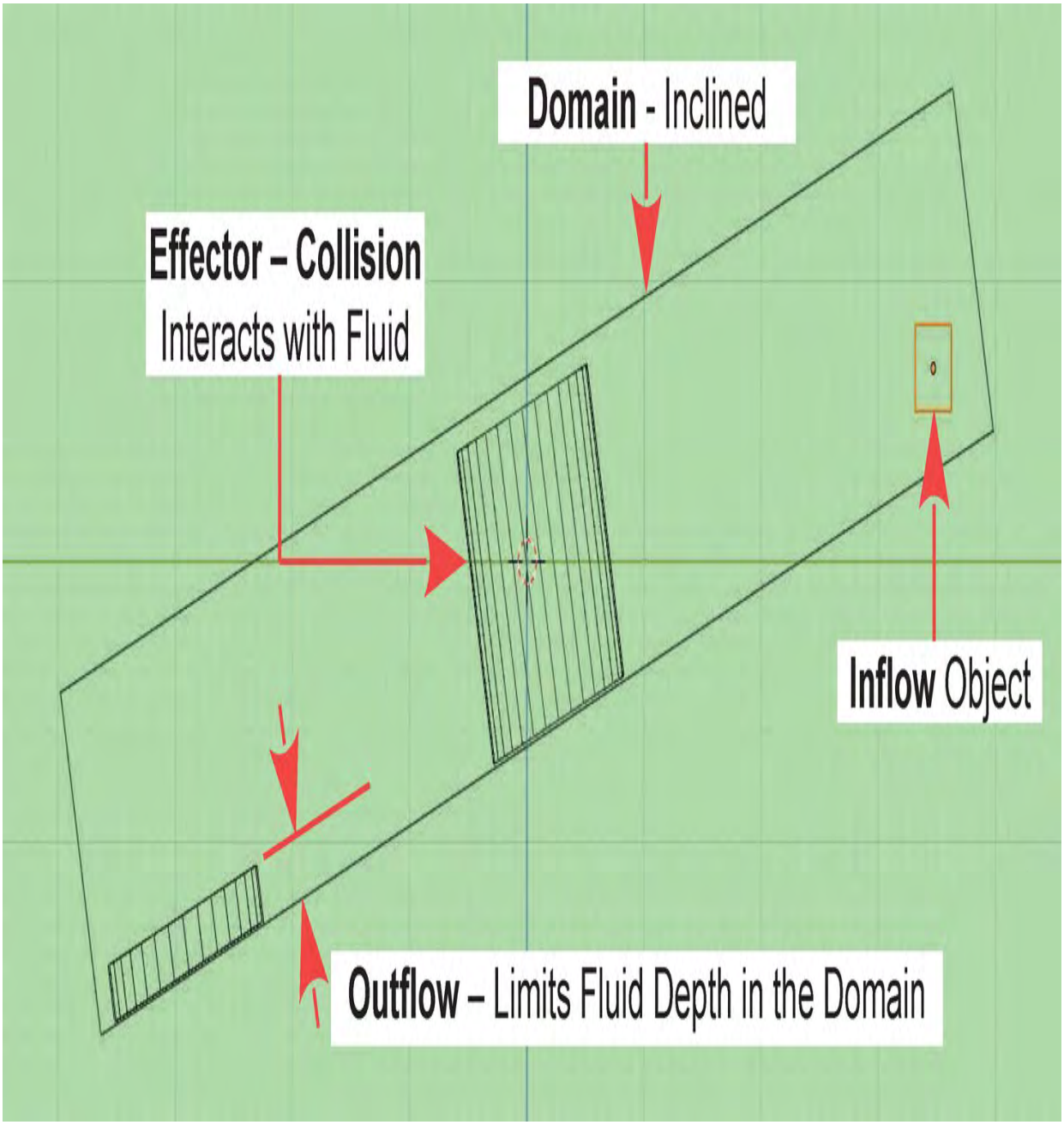
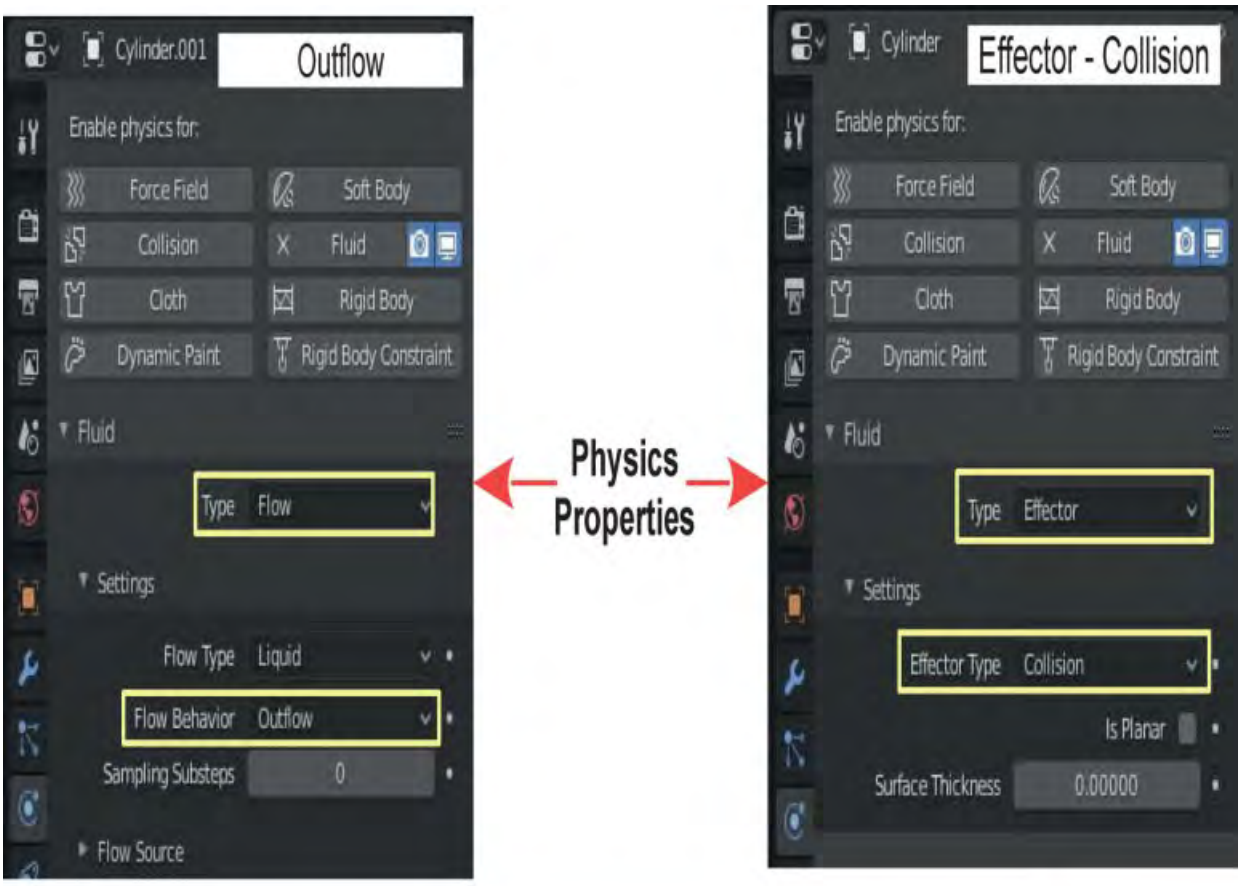


Figure 25.55


Cylinder.001
Outflow
Enable physics for:
Force Field
Soft Body
Collision
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Fluid
Type Flow
Settings
Flow Type Liquid
Flow Behavior Outflow
Sampling Substeps 0
Flow Source
Physics Properties
Cylinder
Effector - Collision
Enable physics for:
Force Field
Soft Body
Collision
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Fluid
Type Effector
Settings
Effector Type Collision
Is Planar
Surface Thickness 0.00000

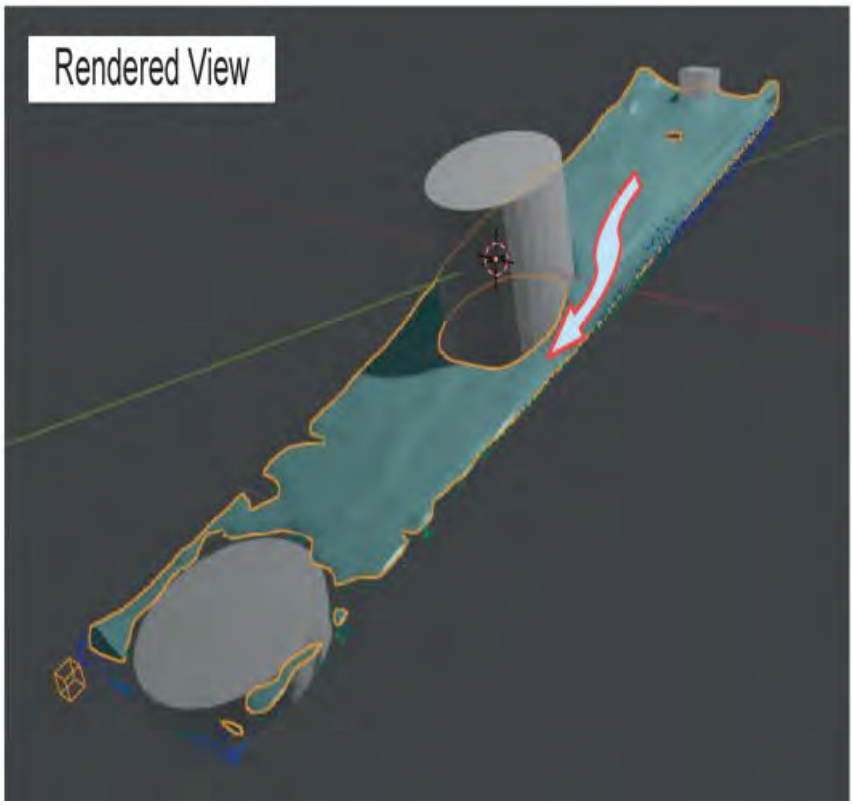
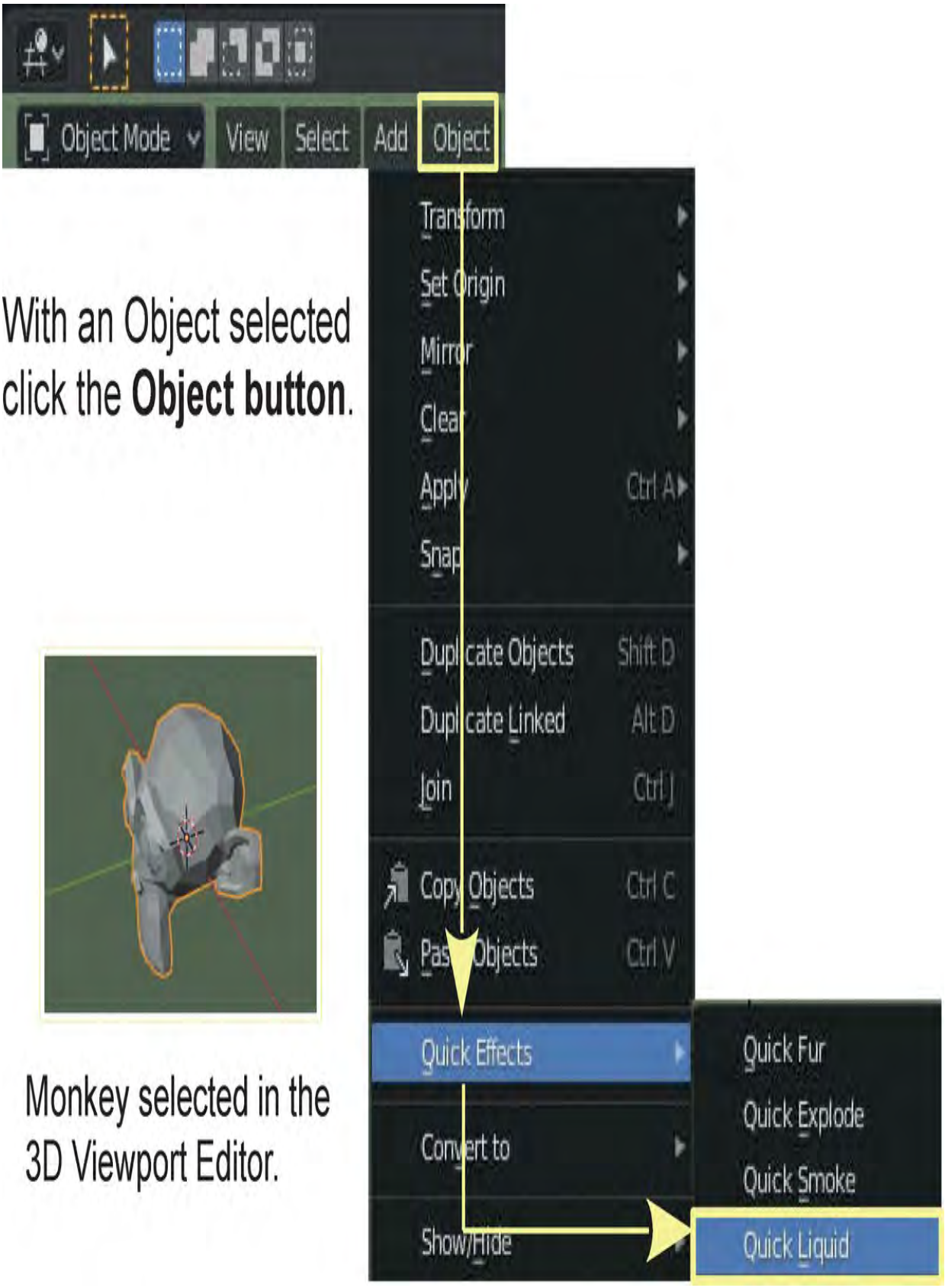

Rendered View

Figure 25.56

25.8 Quick Methods (Reference: Quick Fur – Hair Particles, Quick Explode - Modifiers)



With an Object selected click the **Object button**.

Monkey selected in the 3D Viewport Editor.

Figure 25.57

Quick Methods or **Quick Effects** are shortcuts for creating Simulations. They are activated **with an Object selected in the 3D Viewport Editor** from the **Object button** in the 3D Viewport Editor Header (Figure 25.57).

Introducing Quick Methods at this juncture may appear to be shutting the door after the horse has bolted but activating the replay in Physics Simulation requires a knowledge of Simulation Controls.

It is also advantageous to understand how a Quick Effect works, in a practical sense, so you can use a Quick Effect by adding to it.

Consider a Quick Effect for a Fluid Simulation. The Quick Effect for Fluid Simulation is labelled **Quick Liquid.**

As an example, delete the default Cube in a new Blender Scene and add a Monkey Object. **With the Monkey selected** click on **Quick Liquid** in the Quick Effects menu.

The Monkey is entered in the 3D Viewport Editor with the Editor in Solid Viewport Shading Mode. Activating the Quick Liquid Effect changes the 3D Viewport Editor to Wireframe Display Mode and the Monkey is surrounded by a Cuboid.

A Fluid Simulation has been created with the Cuboid being the **Domain** and the **Monkey is now a Fluid Flow Object, Flow type Liquid with Flow Behaviour: Geometry.**

Controls have been set in the Properties Editor, Physics Properties (Figures 25.59 and 25.60 opposite).

Flow Behaviour **Geometry** sets the Liquid in the Simulation to be delivered as a Mass inside the Domain.

Press the **Play Button** in the **Timeline Editor** to see a volume of fluid drop from the Monkey into the Domain (Figure 25.62).

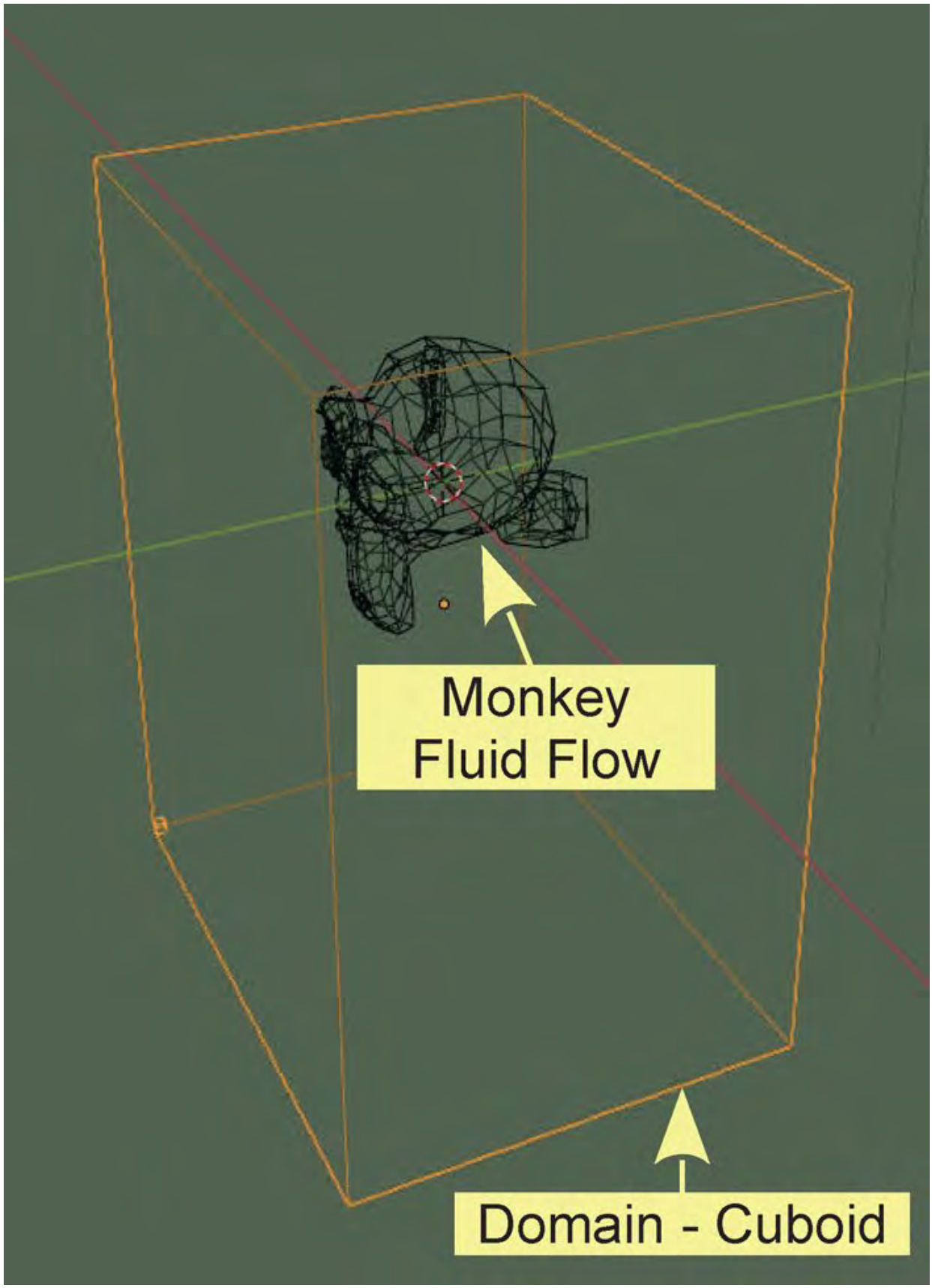
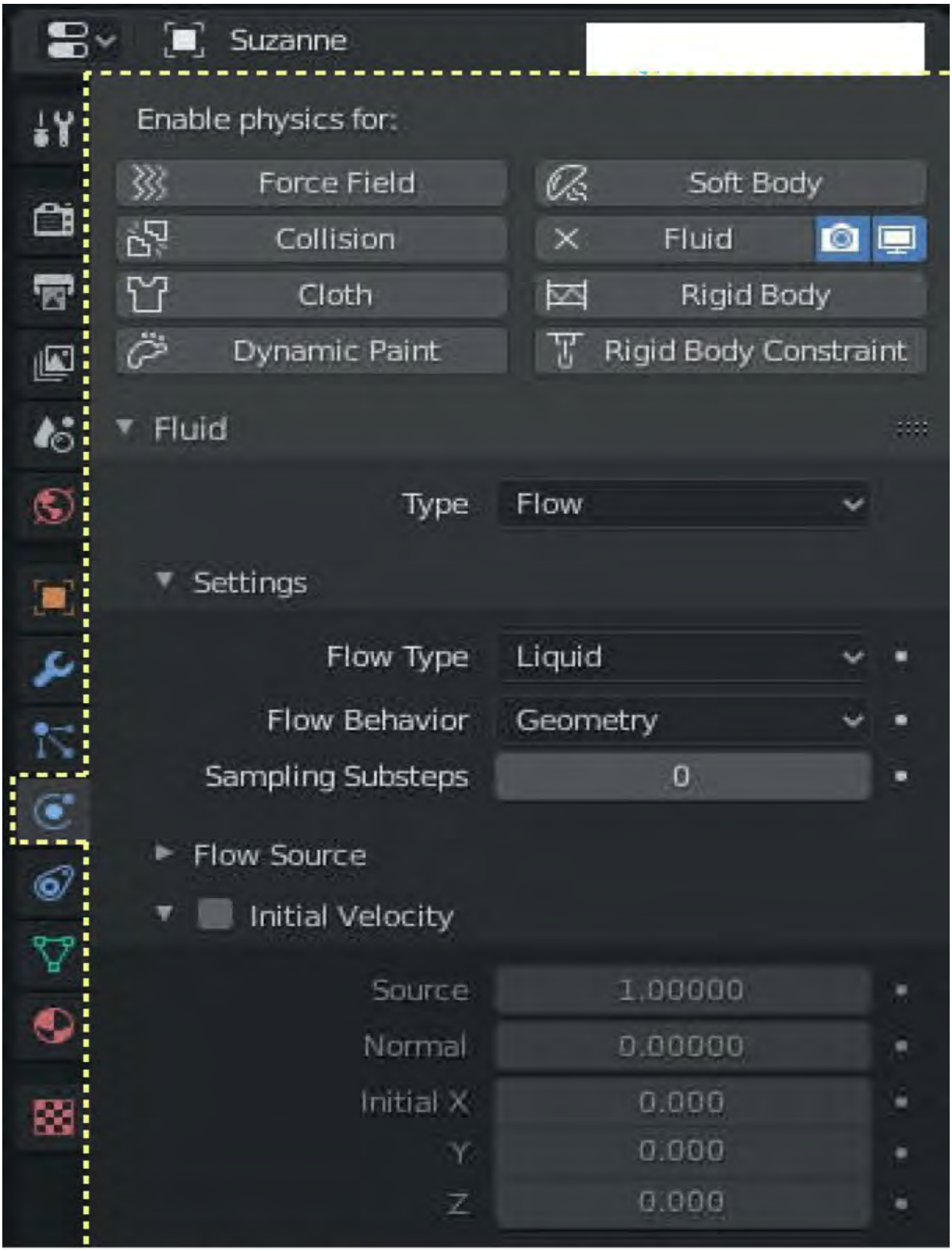
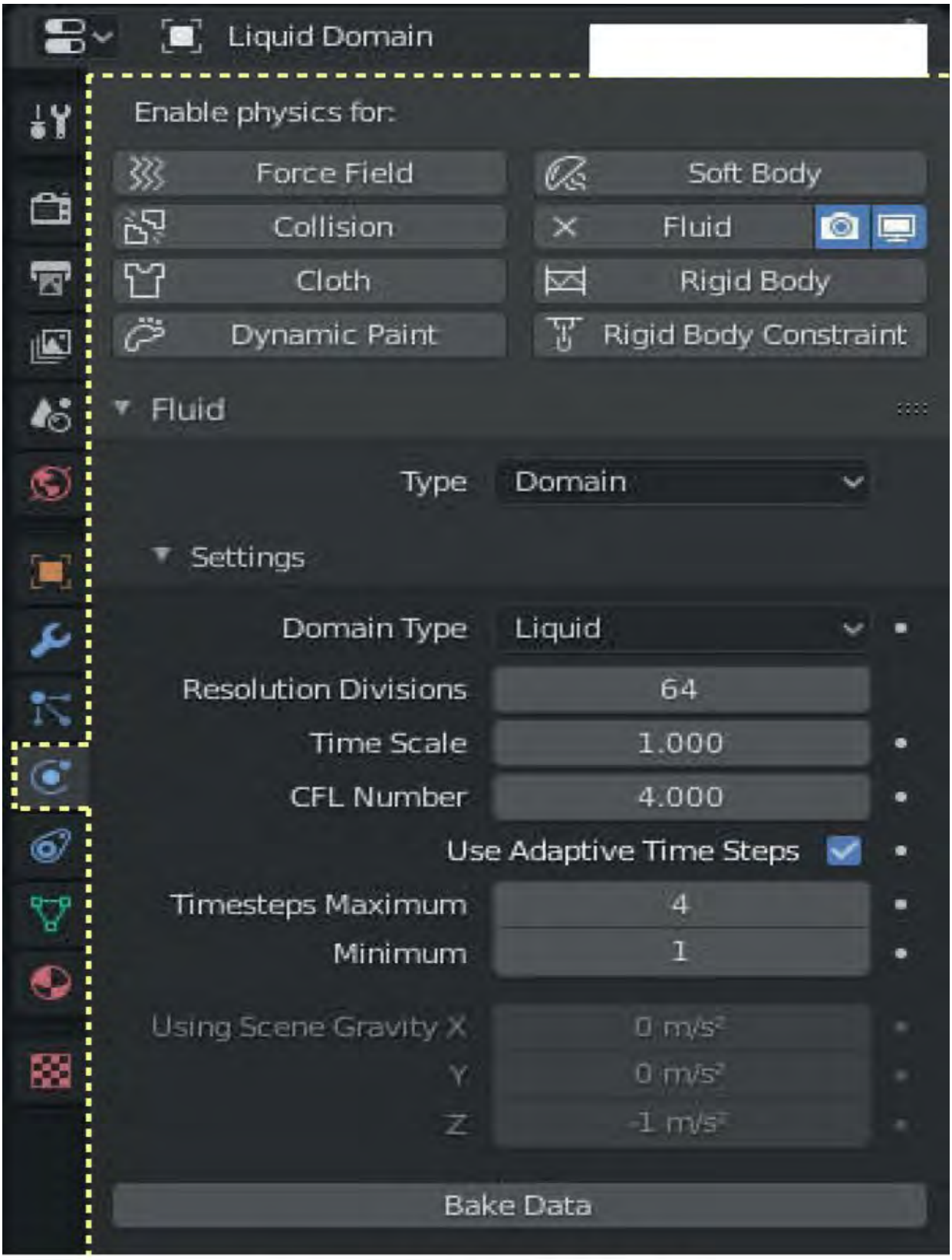

Monkey
Fluid Flow
Domain - Cuboid

Figure 25.58



Monkey – Fluid flow

Figure 25.59



Domain - Cuboid

Figure 25.60

Properties Editor, Physics Properties, Cache Tab.

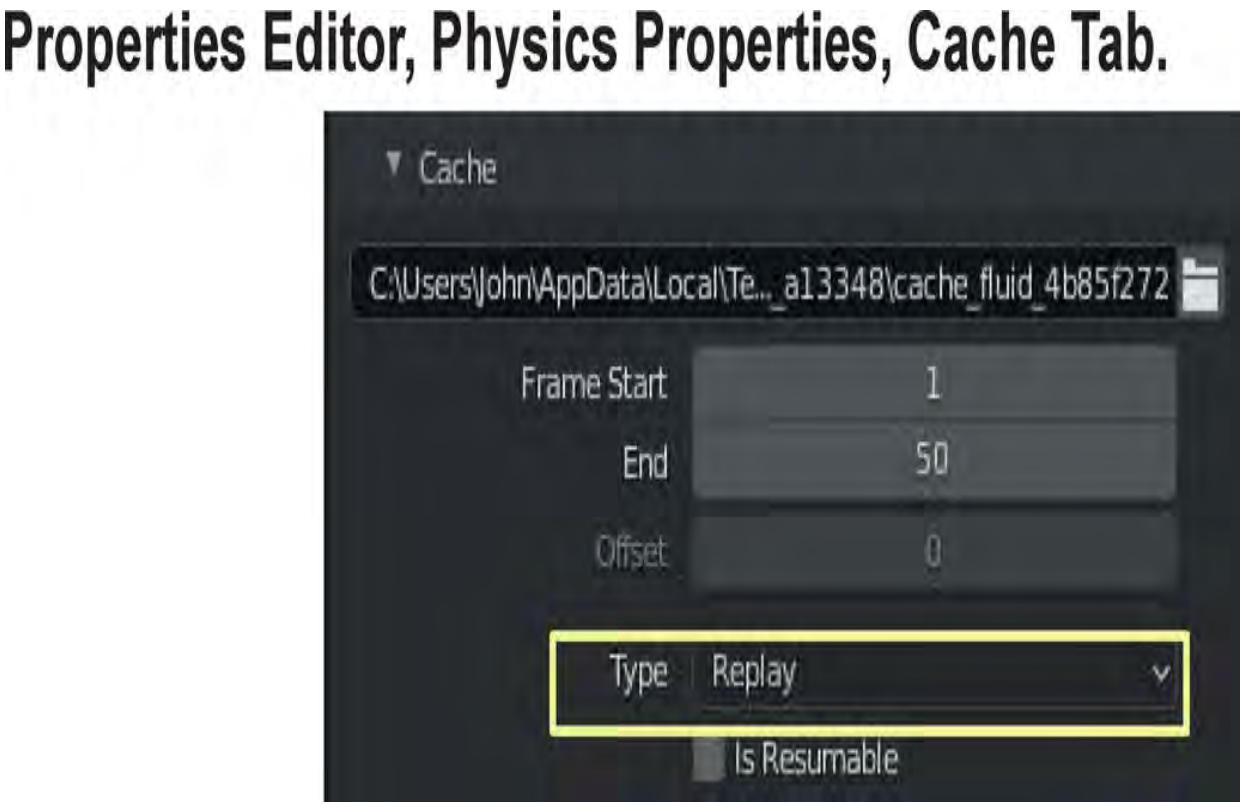


Figure 25.61

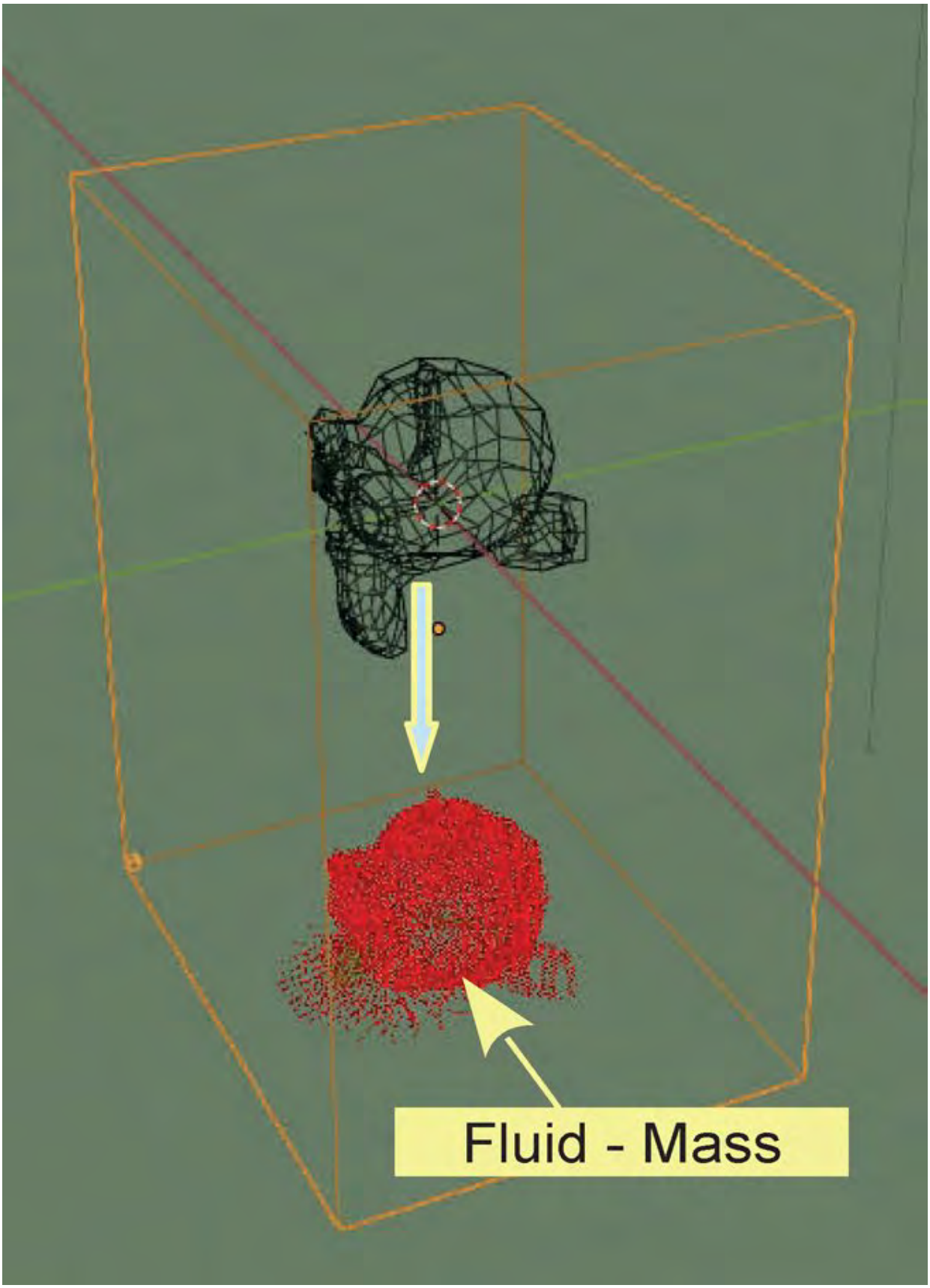

Fluid - Mass

Figure 25.62

Press Play in the Timeline Editor.

> **Note: With Cache Type Replay,** when setting up any Simulation, allows you to Play the Animation and preview the action before performing a Bake. This is very useful in determining if the Simulation will perform correctly.

25.9 Fluid Simulation Continued

The Fluid Simulation setup in section 25.7 has demonstrated a basic configuration using a Fluid Flow Type: **Flow** with Settings, **Flow Type: Liquid** and Flow Behaviour: **Inflow.**

Flow Behaviour: Inflow produces a continuous delivery of Fluid over the length of the animation. To expand on the topic, examine the difference in Fluid Behaviour when **Type Geometry** is employed. Instead of a continuous flow, Type Geometry delivers the Fluid in a mass. An example of this type of delivery was demonstrated by using the **Quick Liquid** method (Section 25.8).

Filling a Cup

In this example **a volume** of fluid will be generated to fall into a cup.

The arrangement in Figure 25.63 shows the 3D Viewport Editor in Wireframe Viewport Shading Mode. The Scene has been constructed with a fluid Emitter (UV Sphere), a Domain Cube (the default Cube scaled up), and an Obstacle Object (a Cup). For the Cup see Chapter 8 – 8.14 and 8.15.

Note: Providing you have saved a Blender file containing the Cup you will be able to append into a new Blender Scene. **Important: When using the Cup developed in Chapter 8, Apply the Modifiers.**

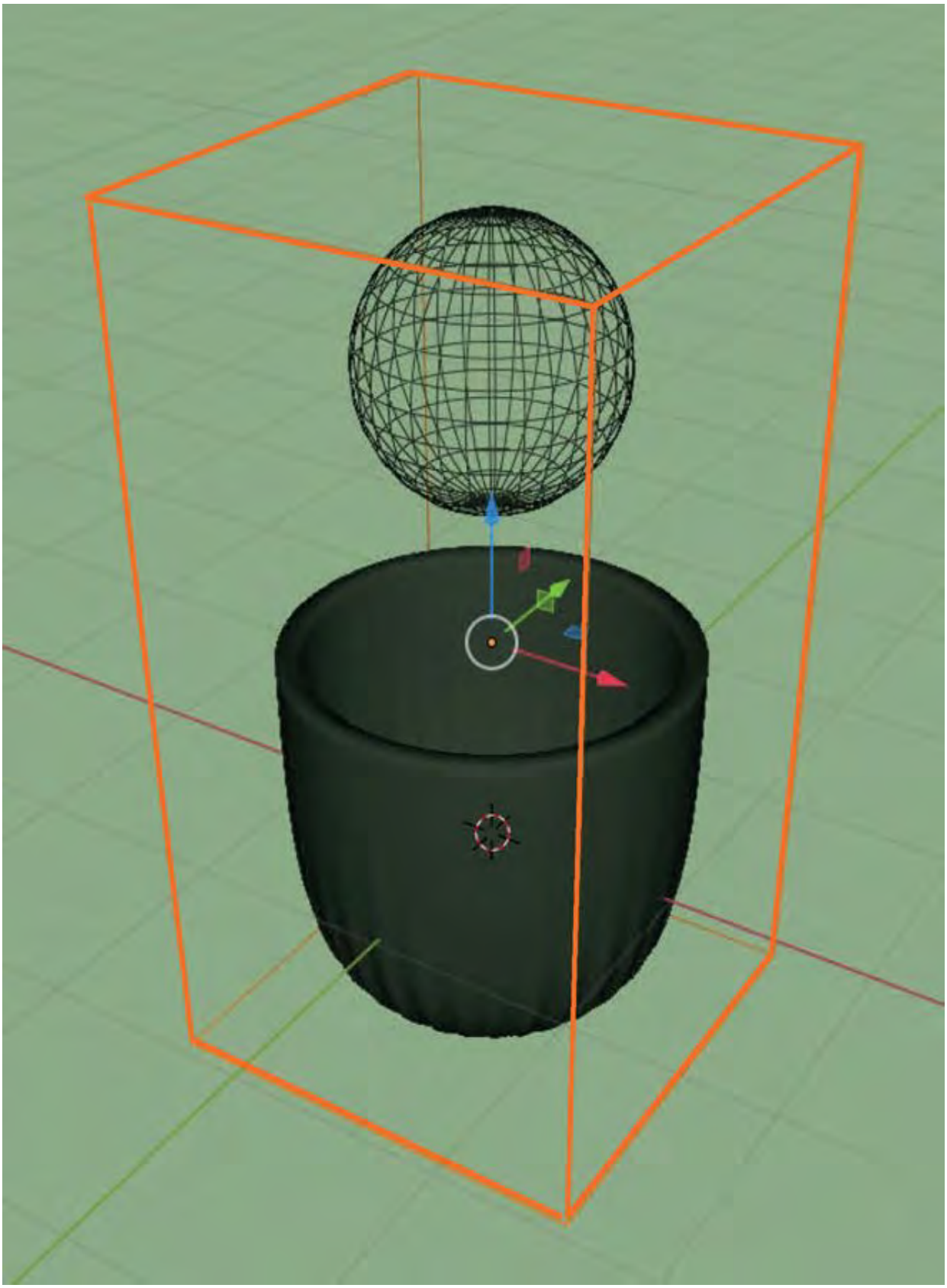


Figure 25.63

Domain Object Setup

The Domain is a Cube that has been scaled to enclose the Sphere and the Cup. For this demonstration default Physics settings will be used.

With the Cube selected, in the 3D Viewport Editor (Figure 25.63). Go to the **Properties Editor, Physics Properties** (Figure 25.64) and click on **Fluid** to display the Fluid Tab. Change the Fluid Type to **Domain.** In the Settings tab set **Domain Type** as **Liquid.**

Scroll down and check **Mesh.**

In the **Cache Tab** change the default **End Frame** to **50.**

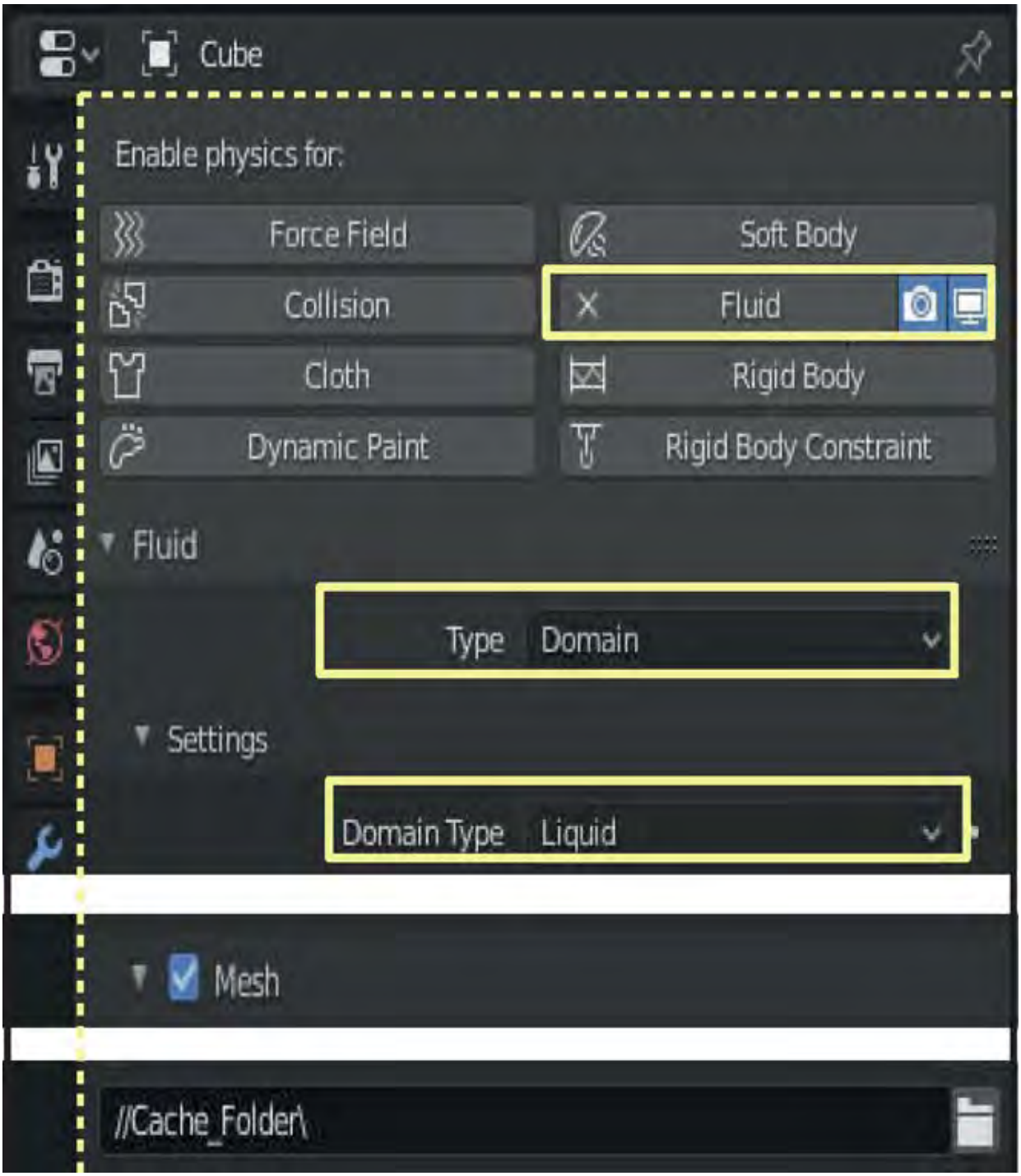


Figure 25.64

Generator Object Setup

The fluid will be emitted by the Sphere that has been placed in the Domain immediately above the Cup. The convention of calling this type of Object the

Generator is adopted.

It is tempting to call the Sphere simply the Fluid Object, since it controls the fluid generation, but all Objects included in the simulation which have Fluid Physics applied, are Fluid Objects.

Select the UV Sphere. In the **Properties Editor, Physics Properties** set values as shown in Figure 25.66. Flow Behaviour: **Geometry** sets the Fluid inflow to take the form of the Generator Object. This means the Fluid will be generated as a spherical mass rather than the continuous flow seen in Section 25.7.

Obstacle Object Setup

In the **Timeline Editor** set the **End Frame** to **50** (Figure 25.65).

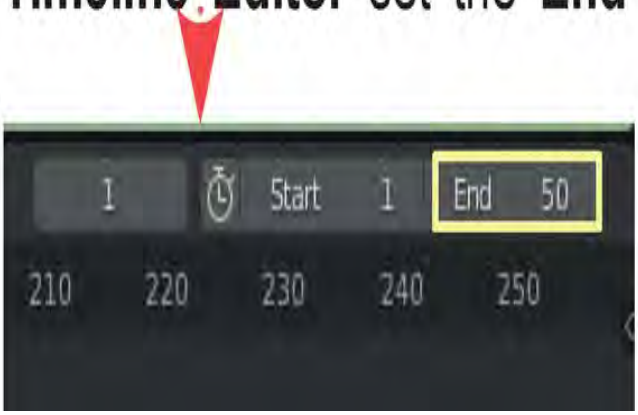


Figure 25.65

Objects included in the simulation which interact with the fluid are called **Effectors.** The Obstacle to the fluid flow inside the Domain will be the Cup.

Select the Cup. Set values as shown in Figure 25.67. Note the Effector Object is named **Cylinder** in the Outliner Editor since the Cup Model has been constructed from a default Cylinder Object and hasn't been renamed. Note also, the **Effector Type: Collision.**

Scale and Proportion

When creating a Fluid animation you should consider the proportion (size relationship) of Objects. In this particular simulation the Sphere Flow Object has been scaled relative to the Effector Cup. The volume of the Sphere determines the volume of Fluid generated, therefore, it has been scaled to produce a volume of Fluid that will reasonably fit into the Cup.

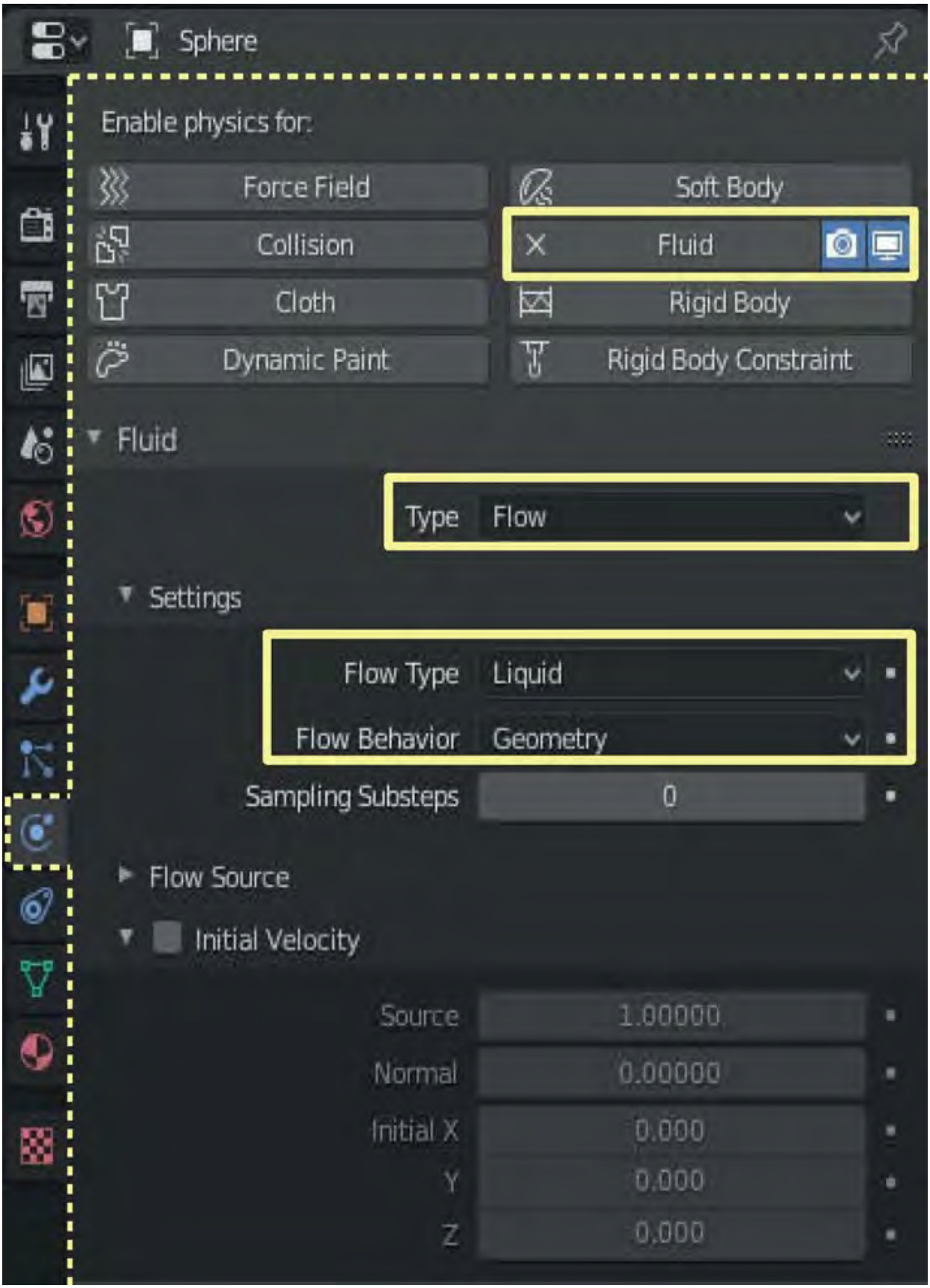

Sphere
Enable physics for:
Force Field
Soft Body
Collision
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Fluid
Type
Flow
Settings
Flow Type
Liquid
Flow Behavior
Geometry
Sampling Substeps
0
Flow Source
Initial Velocity
Source
1.00000
Normal
0.00000
Initial X
0.000
Y
0.000
Z
0.000

Figure 25.66

As it is, the Cup is 1.99m in diameter which is a very large Cup. If this were to be placed in a Scene with other Objects the scale of everything would have to be relative.

Another factor to consider is Time. Look at the arrangement and think about how long you want this huge drop of Fluid to take to drop into the Cup. So that you can see what happens when this occurs, perhaps two seconds is appropriate.

The animation length has been set at 50 Frames. The output Frame Rate, by default, set in the Properties Editor, Output Properties buttons is 24 fps (Frames Per Second).

50 Divided By 24 = 2.08333 seconds

If you want exactly Two seconds change the Frame Rate to 25.

Viscosity

Yet another factor to consider is Viscosity. The viscosity refers to the "thickness" of the fluid and actually the force needed to move an object of a certain surface area through it at a certain speed. Blender uses the kinematic viscosity which is dynamic viscosity.

The table below gives some examples of fluids together with their dynamic and kinematic viscosities.

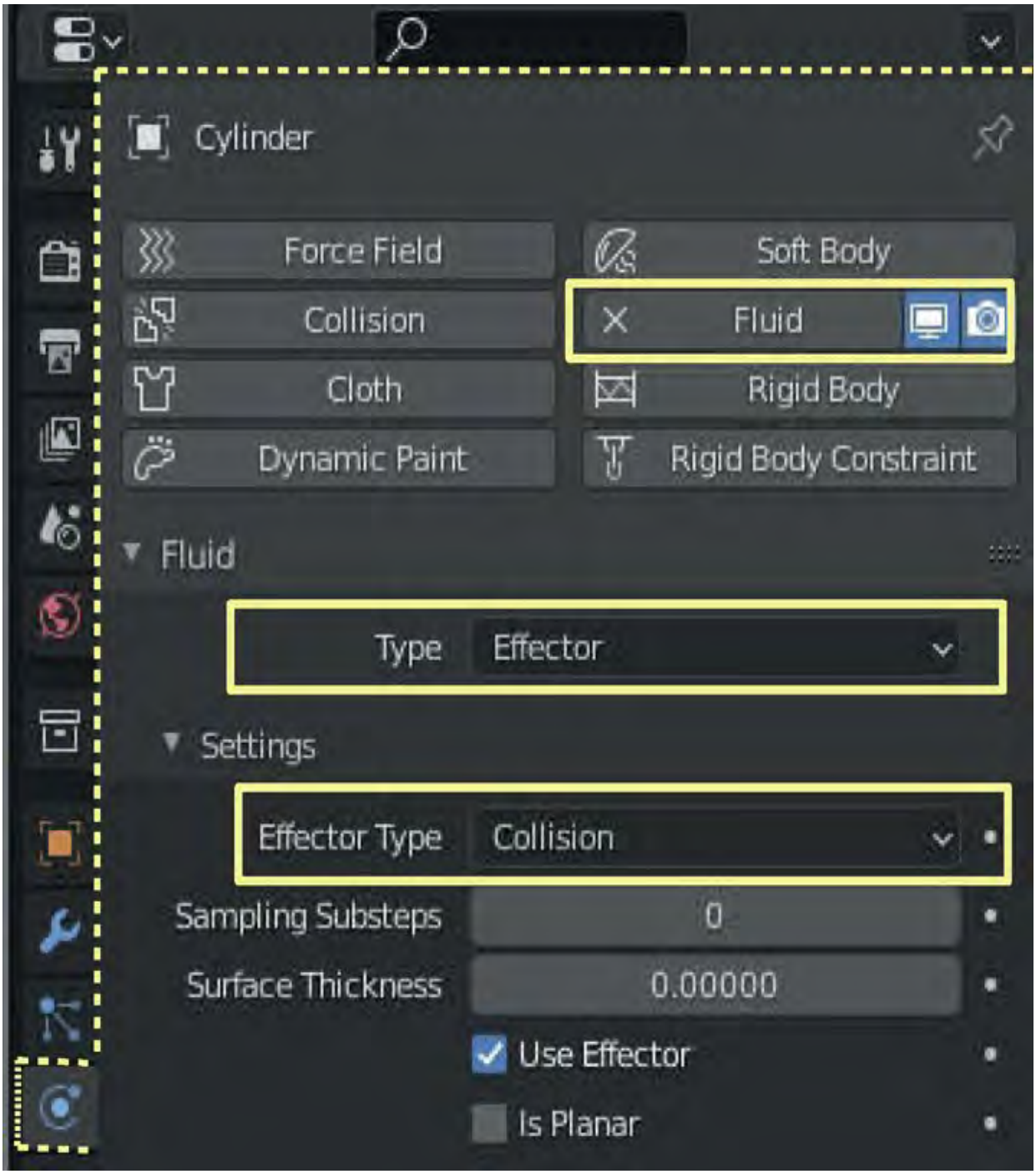

Cylinder
Force Field
Soft Body
Collision
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Fluid
Type
Effector
Settings
Effector Type
Collision
Sampling Substeps
0
Surface Thickness
0.00000
Use Effector
Is Planar

Figure 25.67

Fluid	Dynamic viscosity (in cP)	Kinematic viscosity (Blender, in $m^2.s^{-1}$)
Water (20 °C)	1.002×10^{0} (1.002)	1.002×10^{-6} (0.000001002)
Oil SAE 50	5.0×10^{2} (500)	5.0×10^{-5} (0.00005)
Honey (20 °C)	1.0×10^{4} (10,000)	2.0×10^{-3} (0.002)
Chocolate Syrup	3.0×10^{4} (30,000)	3.0×10^{-3} (0.003)
Ketchup	1.0×10^{5} (100,000)	1.0×10^{-1} (0.1)
Melting Glass	1.0×10^{15}	1.0×10^{0} (1.0)

Blender viscosity unit conversion.

Figure 25.68

Viscosity is entered in the **Properties Editor, Physics Properties,** with the **Domain** selected. The entries are found in the **Diffusion Tab.**

By default the Viscosity for Water is set which is; Base: 1.000 and Exponent: 6.

Note: The Diffusion Tab only displays when the Domain Type is **Liquid** and under Viscosity Presets the default display is **Fluid Presets.** Clicking the Fluid Presets button displays a selection menu for Honey, Oil and Water (Figure 25.69). Selecting an option changes Fluid Presets to the option selected.

With these very basic settings in place, **Bake Data** then **Bake Mesh.** How the Bake performs will depend on your computer power. The fantastic video demonstrations on the internet tend to breeze over this process so be warned and be prepared to wait during Baking. On my PC, for this demonstration, Data Bake and Mesh Bake takes several seconds.

Remember: In the **Cache Tab,** the default Type is Replay. Check **Is Resumable** and change to **Type: Modular. Change the Cache Folder to an empty Folder.**

Bake Data then **Bake Mesh** (expand the Mesh Tab)

When Baking Data is complete the Sphere displays with a blue mesh overlay. When Mesh is Baked the Domain consolidates and takes on the shape of the Sphere.

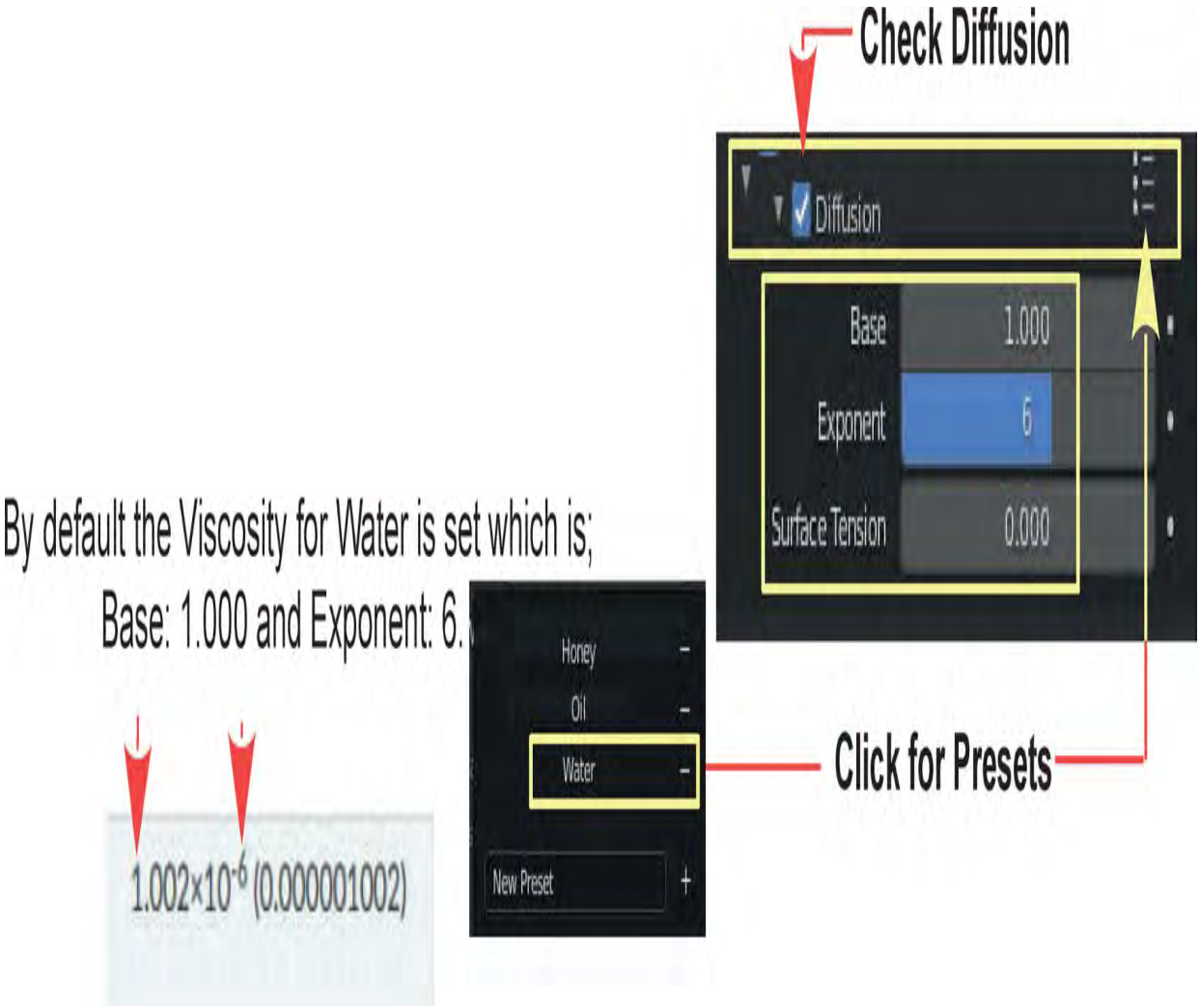


Figure 25.69

Play the Animation in the Timeline Editor.

The consolidated Domain is now the Fluid which descends into the Cup **BUT** as you see it breaks through the Cup Mesh and ends up sitting in what was the profile of the Domain. This is obviously not what you want. A modification is required followed by a Re-bake.

In the 3D Viewport Editor, select the Cup. In the Properties Editor, Physics Properties, Fluid Tab you will see Surface Thickness: 0.00000. This value determines the distance around the Effector Object to be used when calculating the Collision. The default value is: 0.00000

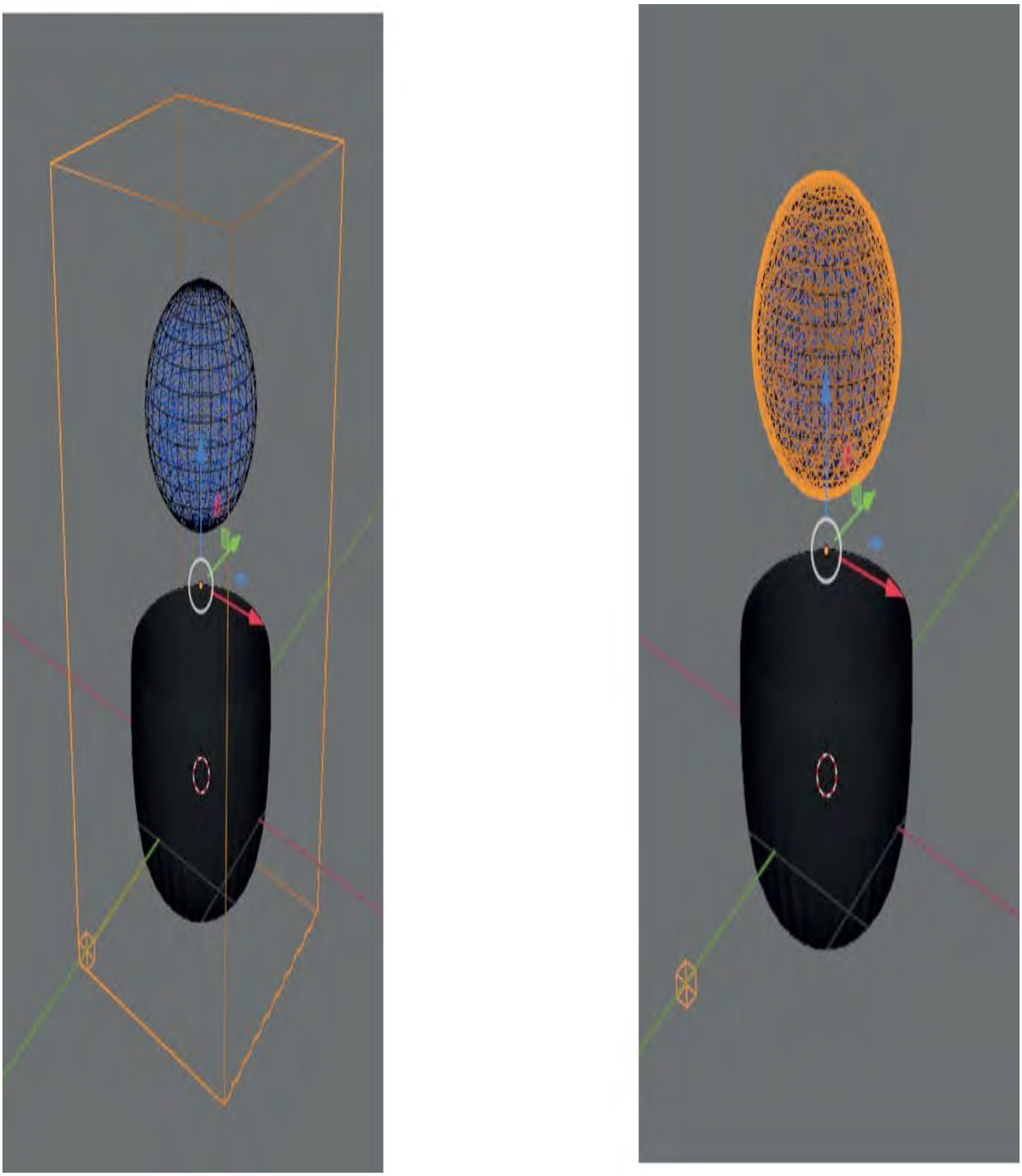


Figure 25.70

A suggestion for a correct value is; 1.0000

Select the Domain Cube in the Outliner which is consolidated as a sphere on the Flow Sphere. Free the Data and Mesh Bakes. Change the Surface

Thickness value, clear the Cache Folder then Rebake Data and Mesh.

Play the Animation to see the Fluid (Mass) descend and fill the Cup (depending on the Sphere size).

Note: If the Bake is not performing as expected, it can be terminated by pressing the **Esc Key** or the Cancel button in the Header. Settings can be adjusted to correct the action. To REBAKE the simulation, select the Domain by clicking on Cube in the Outliner Editor (which is now the blob attached to the sphere) and press Resume to continue the Bake. Alternatively, Free the Bake and Bake a second time.

Note: If the demonstration does not perform as expected you could have made an error in the set up. Check your settings and change values accordingly, BUT Note; having changed a setting you will have to Free Data and REBAKE the simulation. Simply changing settings will not correct the action.

Note: Before setting up a new Fluid Simulation clear the data from the Cache file or set a new location for saving the Bake. If data exists in the Cache when a new Domain is created, Blender will attempt to use the existing data.

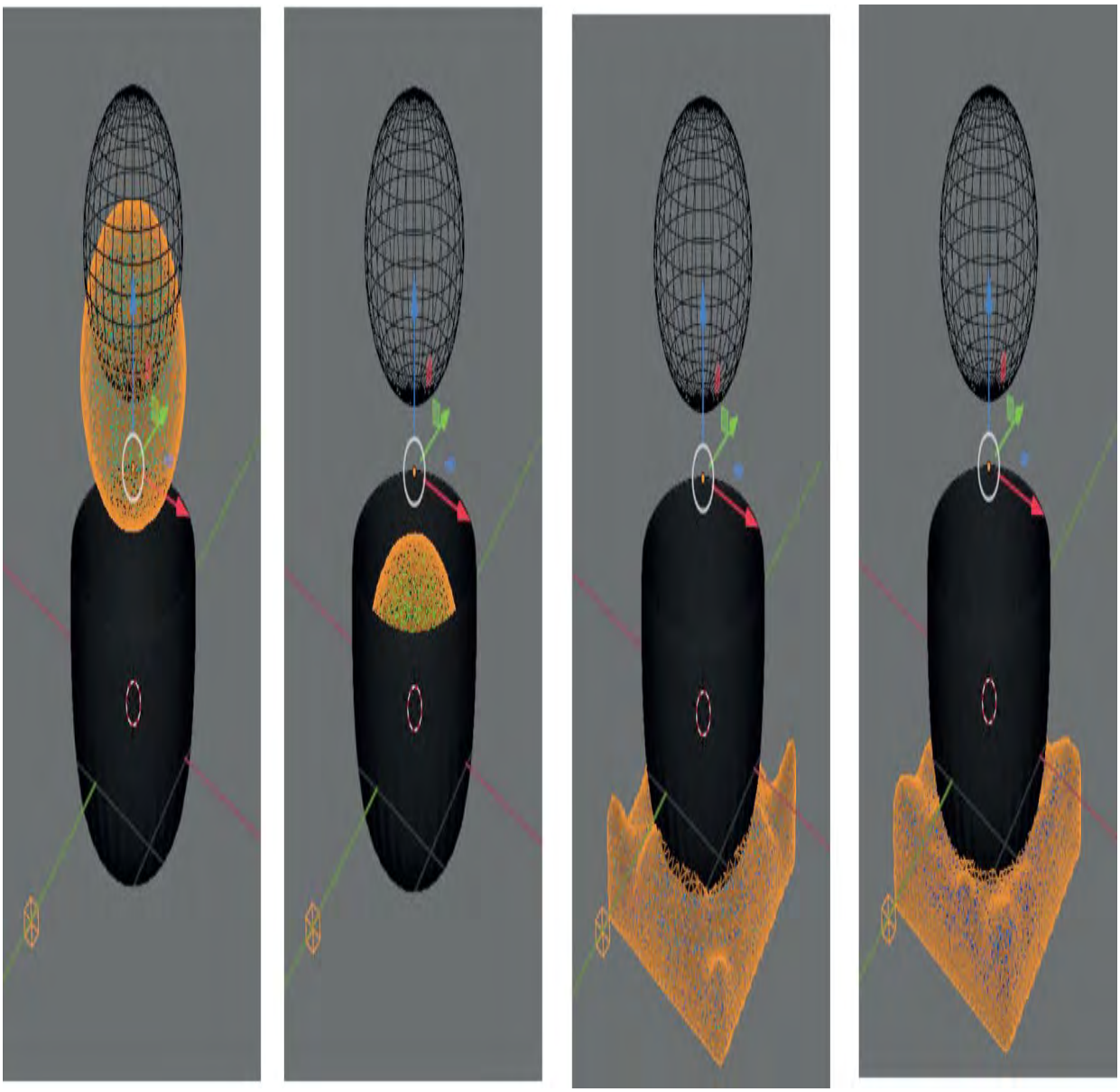


Figure 25.71

25.10 Fluid Simulation Experiment (The experiment will show Fluid filling a Bowl.)

Modeling a Bowl

A simple method of creating a Bowl is the delete the upper part of a UV Sphere Object by selecting Vertices in Edit Mode (Figure 25.74). With the Vertices deleted and with the remaining Vertices selected, press the E Key to

Extrude (duplicates the Vertices) then immediately press the S Key (Scale) and move the Mouse Cursor towards the center, scaling the Extrusion down, forming a inner surface for the Bowl. Finally move the inner selection up to align the rim.

The Bowl will be an **Effector Object** in the Fluid Simulation and as such must have thickness or be a solid Object. An Object such as a Plane will not work as an Effector.

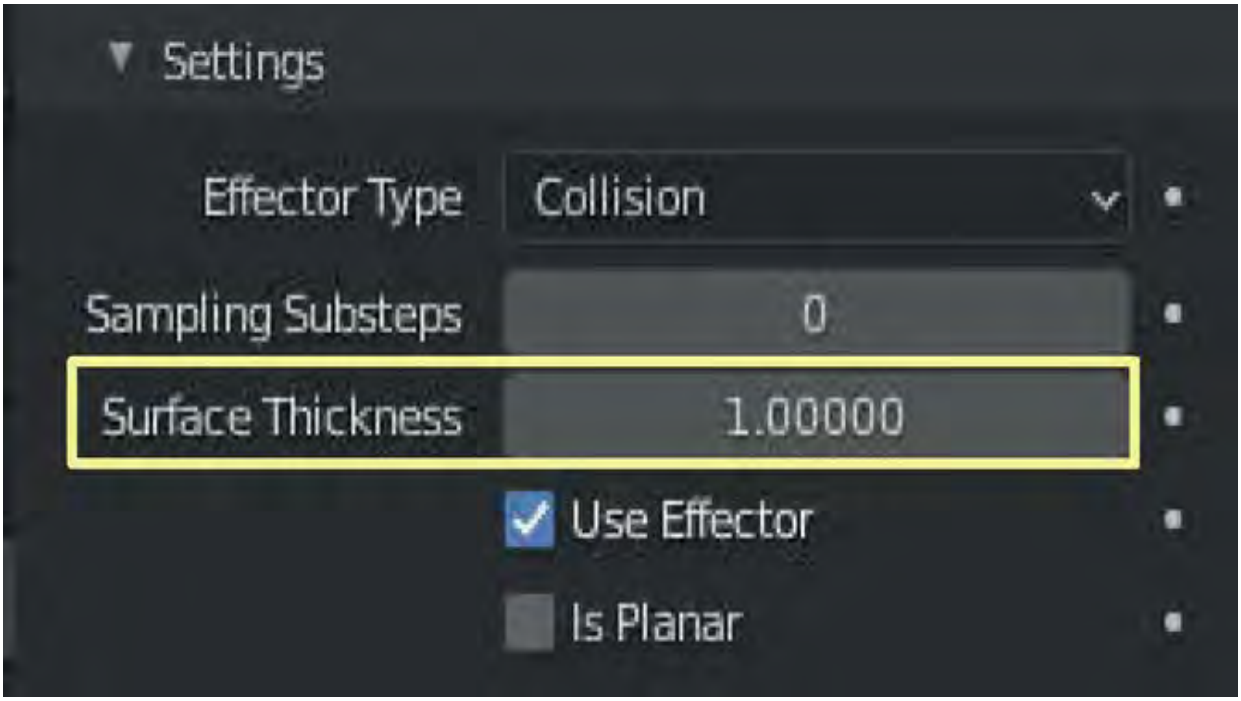


Figure 25.72

Arrange the model of the Bowl inside a Cube Domain and have a smaller second Cube positioned above the Bowl (Figure 25.75). The smaller Cube will be the Fluid Inflow Object.

Fluid Physics Setup

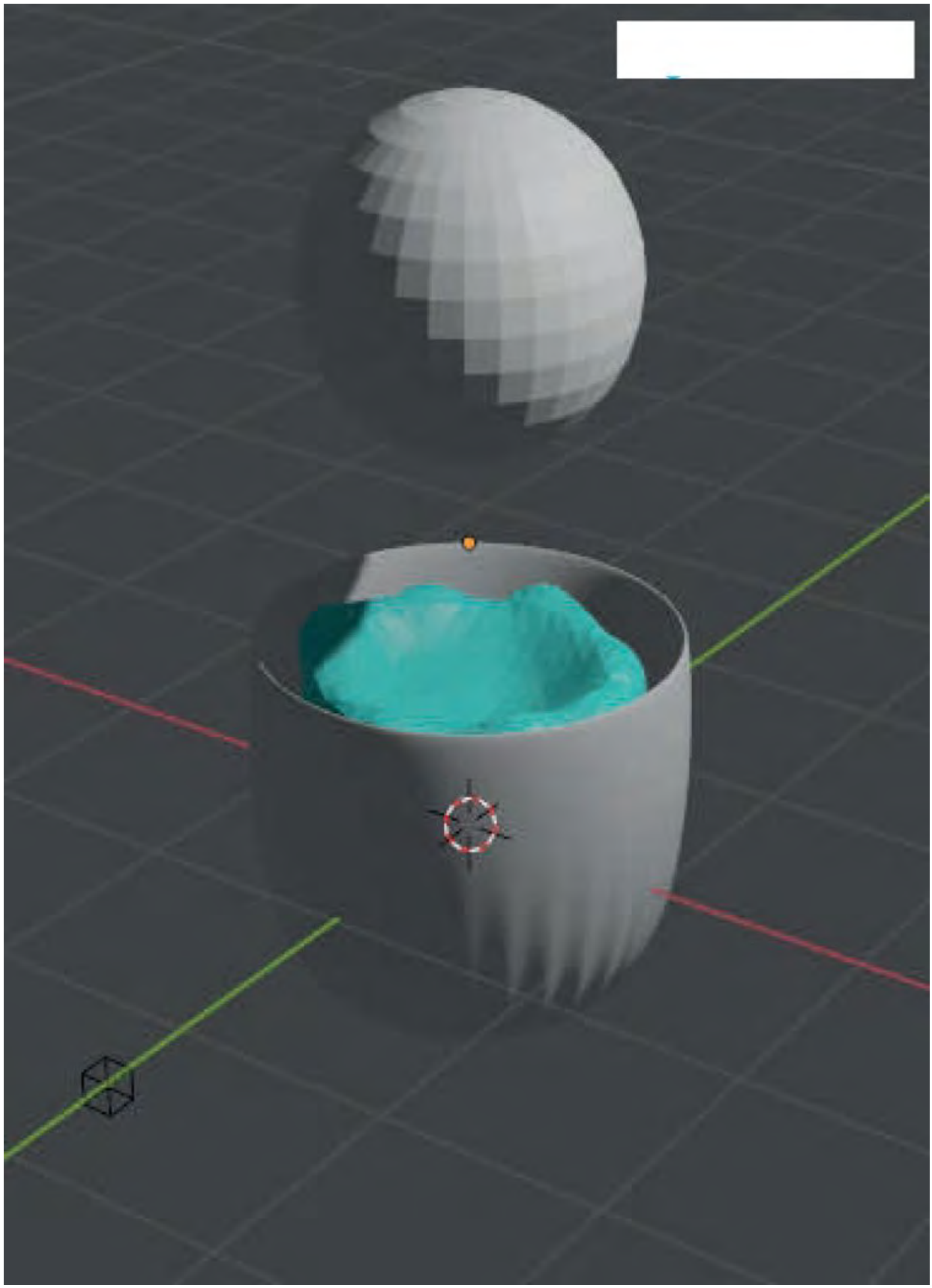


Figure 25.73

Select each Object to be included in the simulation and apply the following settings in the Properties Editor, Physics Buttons.

Domain: Fluid – Fluid Type: Domain – Settings Domain Type: Liquid.

Have the Liquid tab checked.

Have the Mesh tab checked.

Enter the File Path to the Cache Folder.

Emitter: Fluid – Fluid Type: Flow – Flow Type: Liquid – Flow Behaviour: Geometry. (Flow Behaviour Geometry delivers the Fluid as a Mass)

Effector: Fluid – Fluid Type: Effector- Effector Type: Collision.

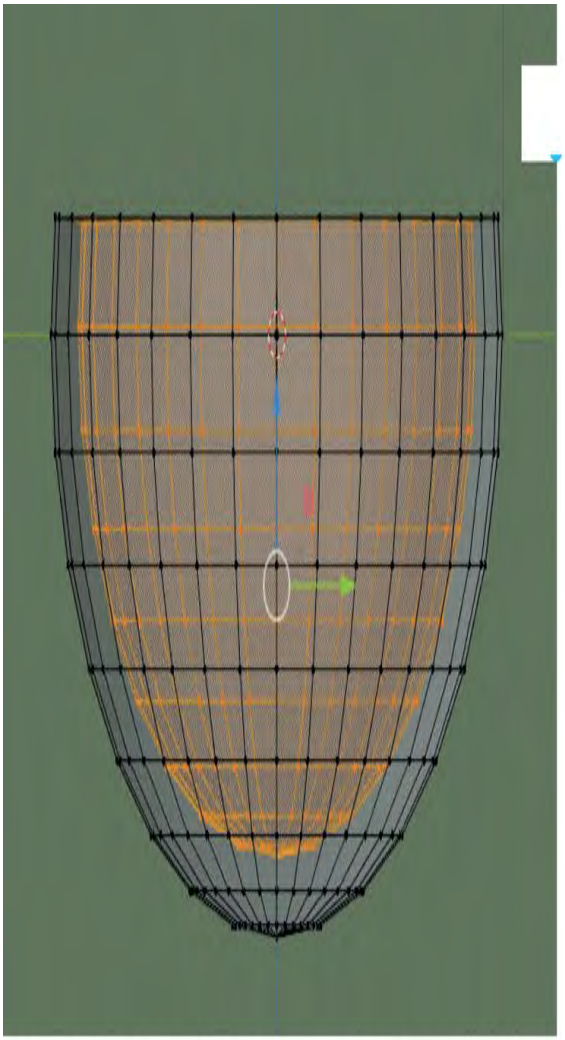
Note: Apart from the above settings default settings are employed.

Timeline Editor: In the Timeline Editor set the Animation End Frame to 50 to correspond to the end Frame value in the Properties Editor, Cache tab for the Domain settings.

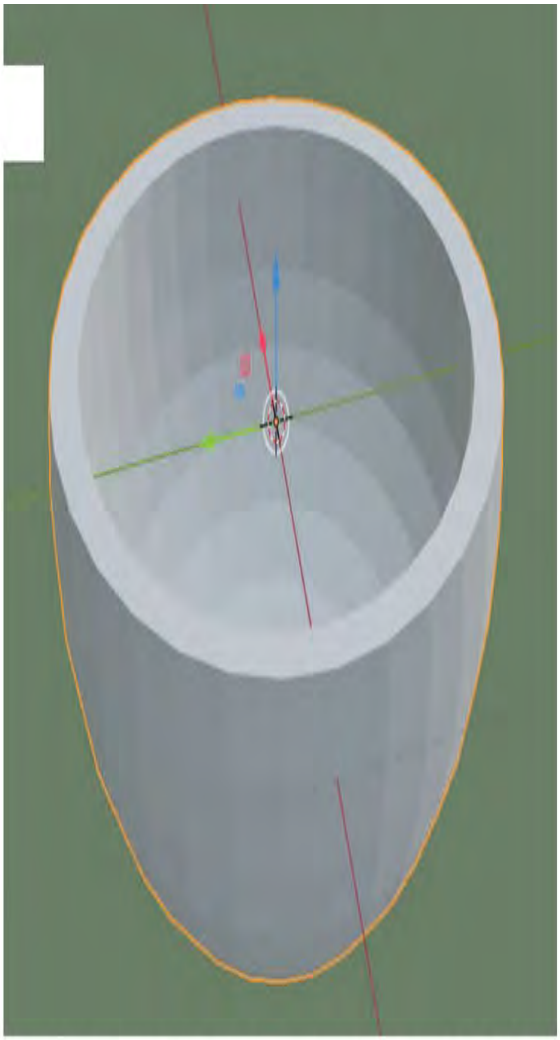
With the above in place, **Bake Data.** Press Bake in the Properties Editor, Physics Properties with the Domain selected.

It is not necessary to Bake Mesh at this juncture. You can play the animation and see the Particles generated to check the Flow in the Domain.

When you play the animation or step through the animation frames in the Timeline Editor, in all probability, you will see the Fluid Particles generated by the Emitter descend and pass straight through the Bowl splashing into the domain (Figure 25.76.



Extruded Vertices aligned at the Rim



Solid Bowl – Solid Viewport Shading Mode

Figure 25.74

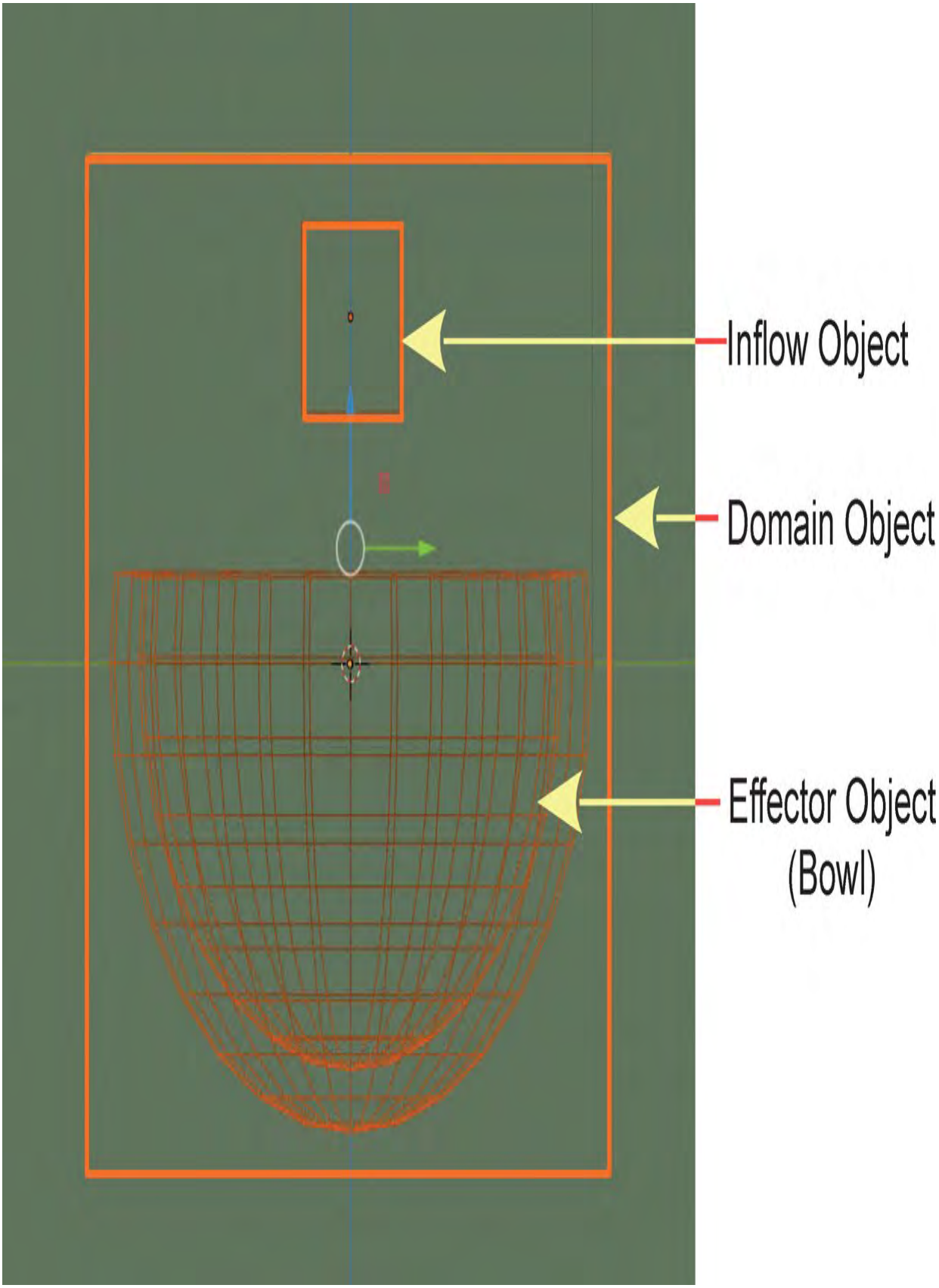

Inflow Object
Domain Object
Effector Object
(Bowl)

Figure 25.75

To correct the Flow, with the Domain selected press **Clear Data** in the Physics Properties to delete the Data in the Cache Folder. In the 3D Viewport Editor, select the Bowl and change the **Surface Thickness** to **1.00000** in the Properties Editor Fluid, Settings Tab.

Note: The correction instructions are purposely repeated to reinforce the concept.

Select the Domain and Rebake Data. Replay the Animation (Figure 25.77). With the correct Fluid Flow, **Bake Mesh** to see the Fluid as a Mesh with the 3D Viewport Editor in Material Preview or Rendered Viewport Shading Mode.

The experiment has been using a **Fluid flow Type Liquid** with **Flow Behaviour: Geometry applied to the Emitter Cube.**

With the identical arrangement, clear the Data to remove data from the Cache Folder then change the **Flow Behaviour** to **Inflow.** Changing to Inflow will deliver the Fluid in a continuous stream instead of a Mass (Figure 25.78). The stream will descend into the Bowl and continuously fill the bowl, eventually overflowing and filling the Domain (Figure 25.79).

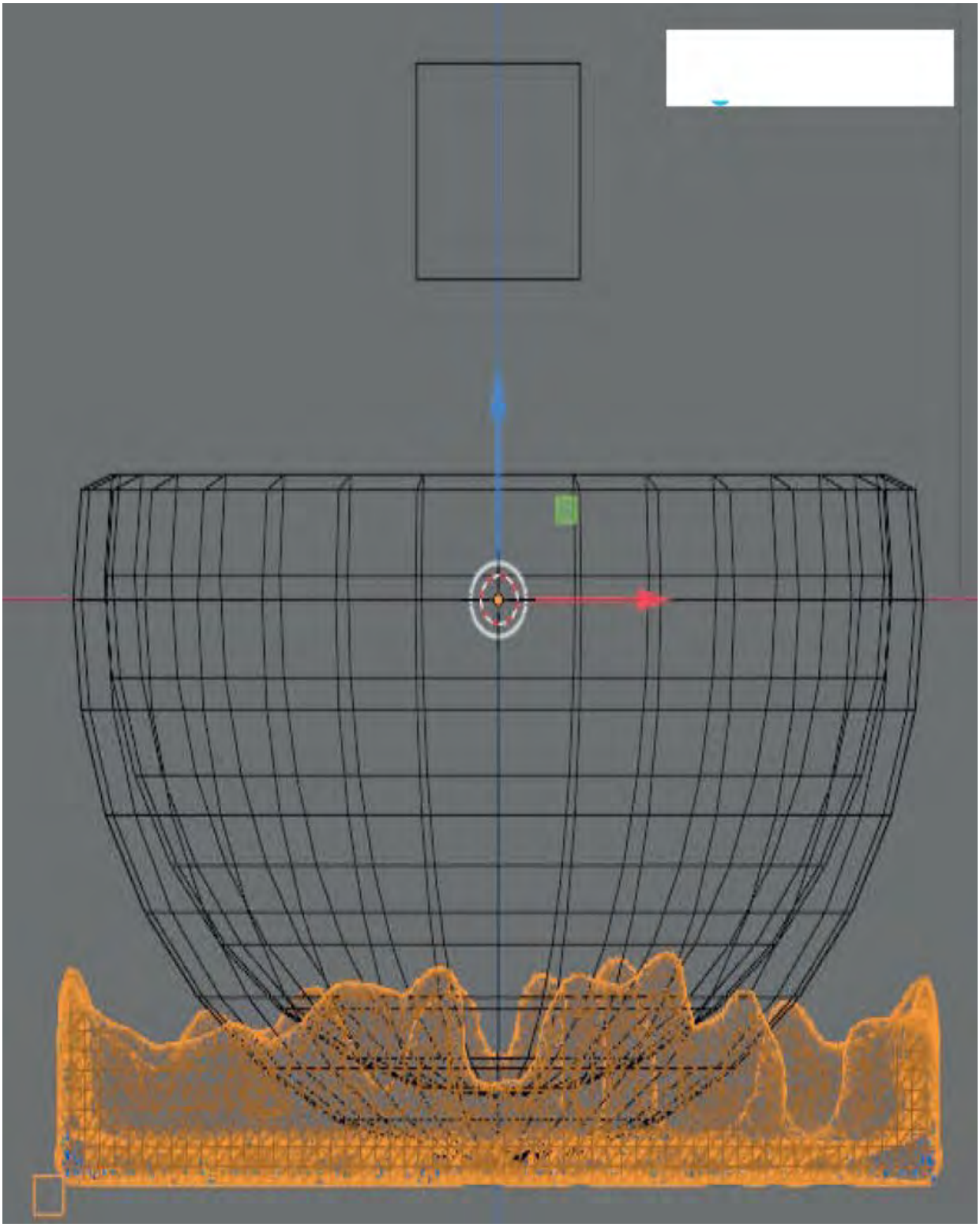


Figure 25.76

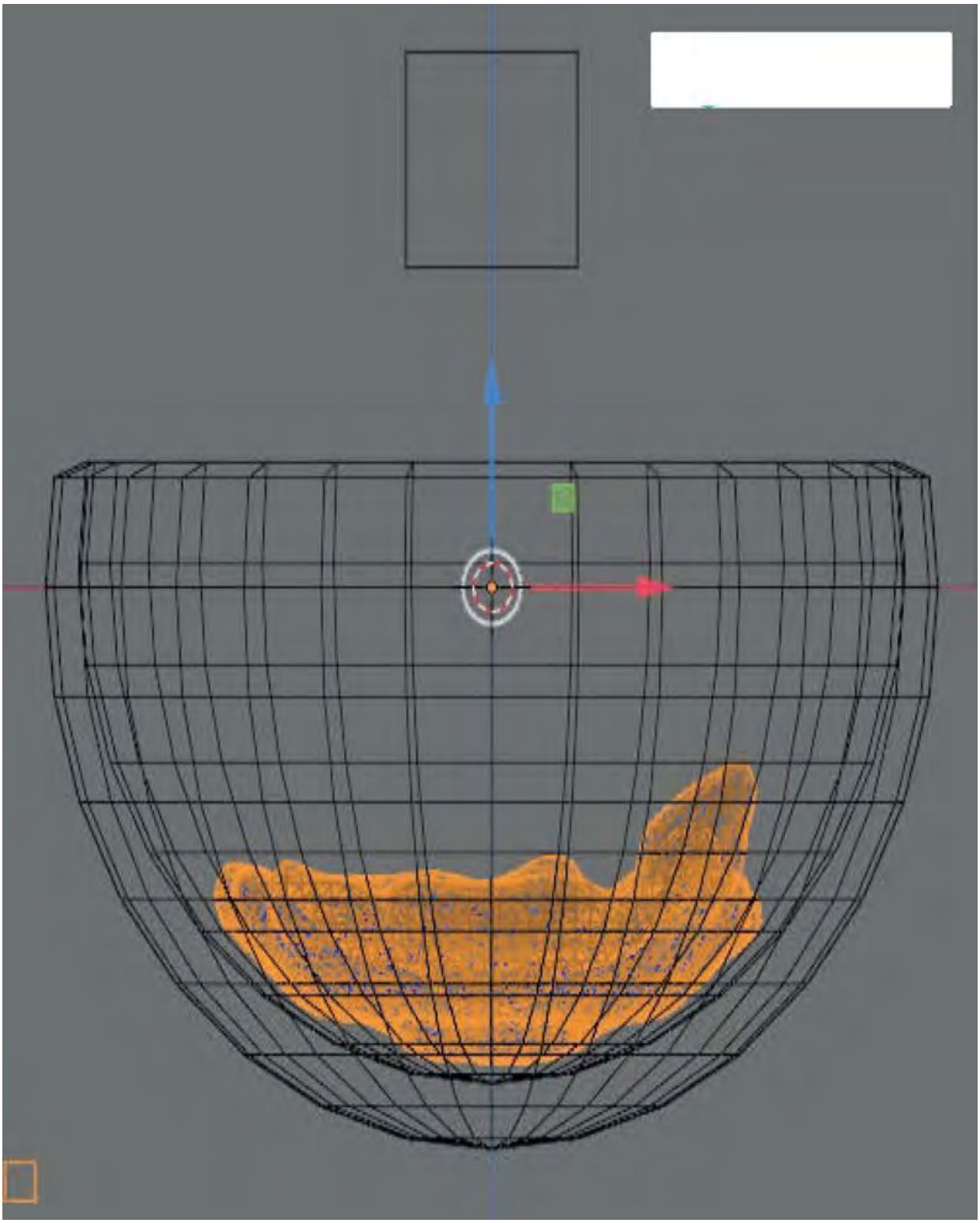


Figure 25.77

With Fluid Simulation the effects are limitless. To discover the possibilities you will have research tutorials on the internet and experiment by yourself. Some effects will be fairly obvious but when you have an idea to create an animation, in the beginning you may find the techniques required to perform the operations required are allusive. The following example will show another variation on what has already been describe.

Ball Dropping into a Cup

This example will demonstrate how to set up an Object (Sphere) to drop into a Cup containing a Liquid. The first task is to model a Cup. This new Cup Model will allow an exact volume of Fluid to just fill the Cup. Start with a **Cylinder Object** and tab into Edit Mode. In Edit Mode change Vertex Select Mode to Face Select Select Mode and delete the top Face of the Cylinder.

Change back to Vertex Select Mode, select the upper rim of Vertices and Scale out slightly forming a thin walled Cup. The Cup will become an **Effector Type Collision Object** in the Fluid Simulation. **Remember;** simple planer objects will not work as Effectors.

To give the Cup wall thickness, select all Vertices, Press the E Key (Extrudes – Duplicates selected Vertices), press the S Key (Scale) and move the Mouse Cursor towards the center, scaling the selected (duplicated) Vertices. While remaining selected move the selection up forming a flat rim (Figure 25.80 over).

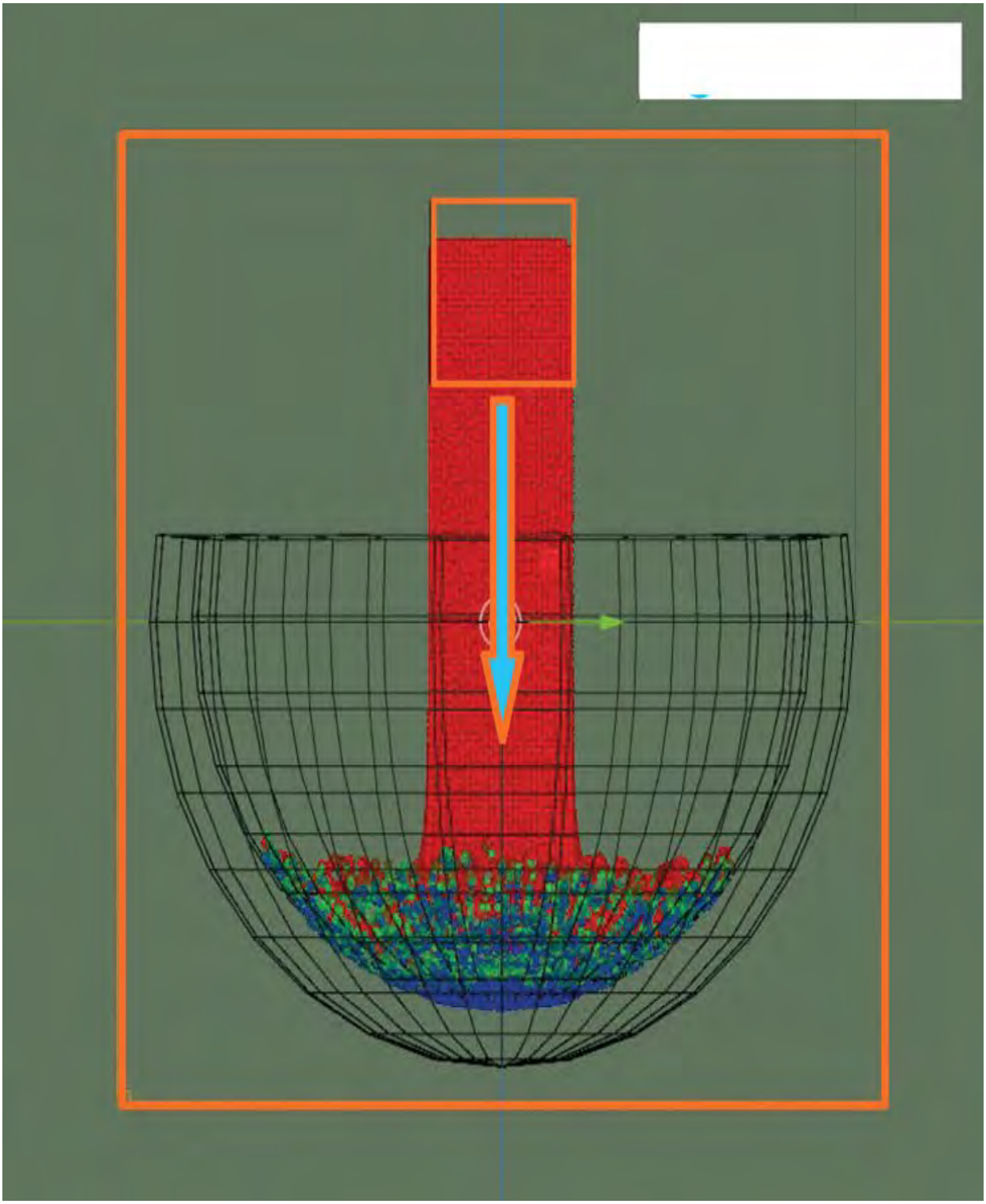


Figure 25.78

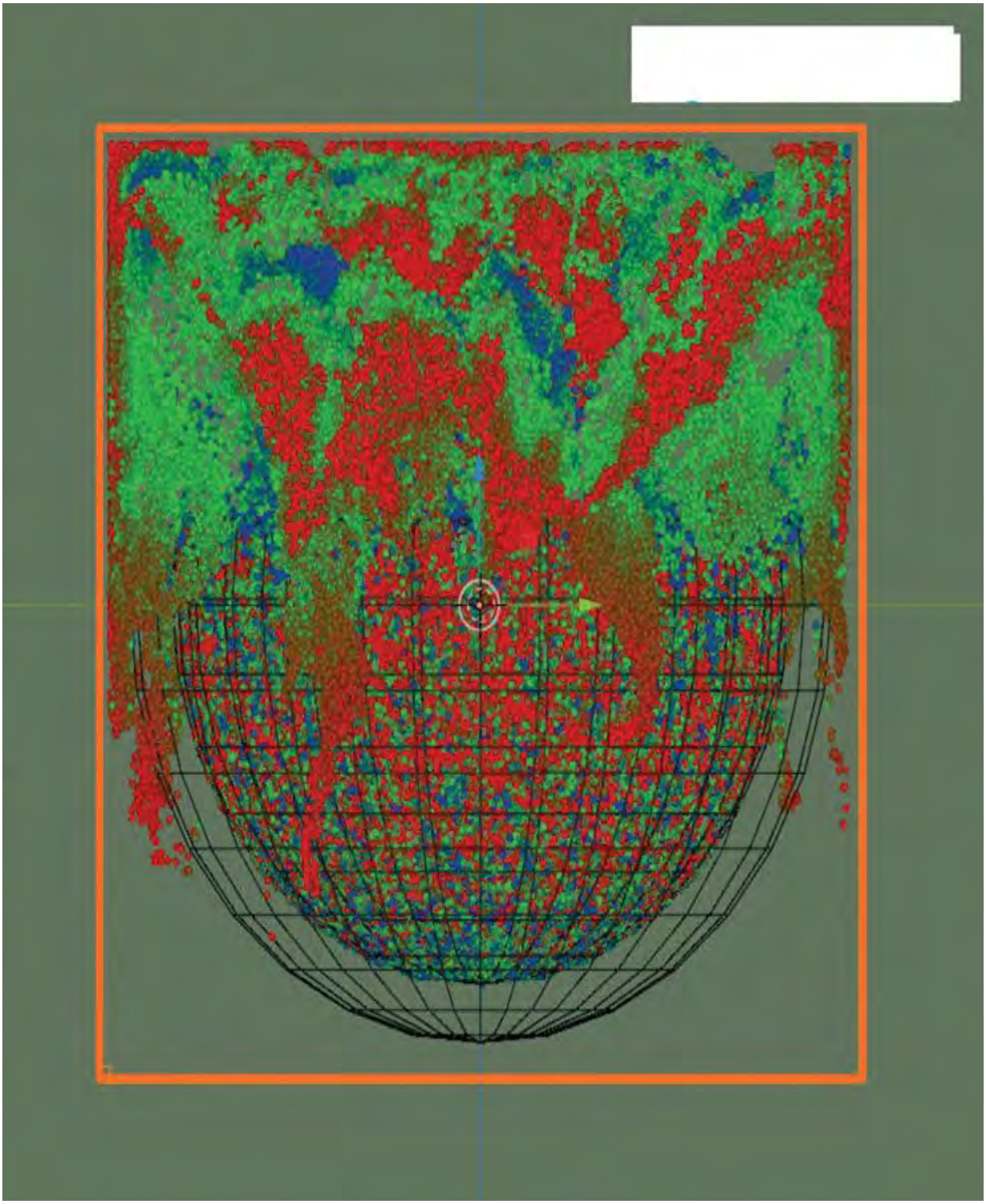


Figure 25.79

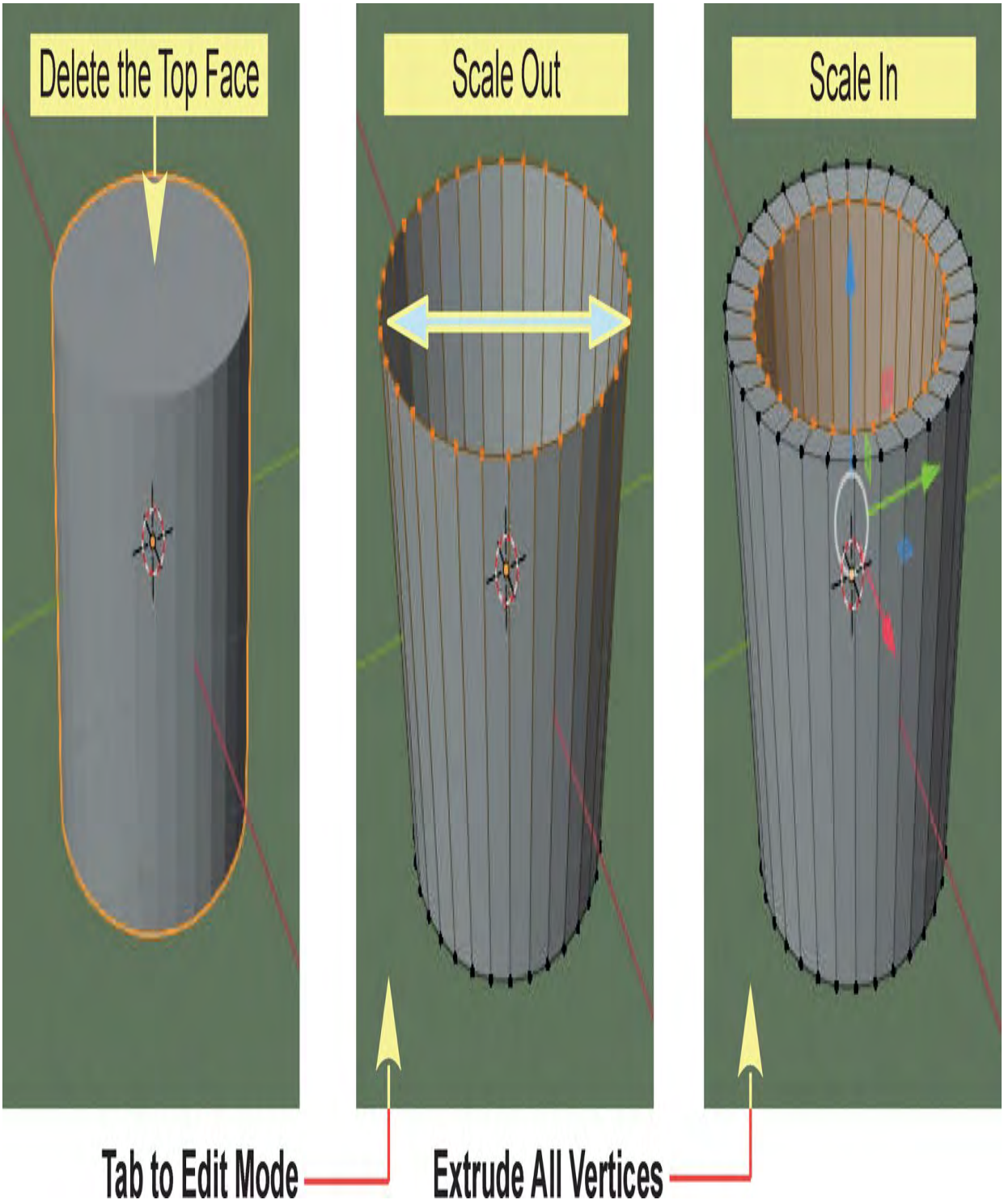
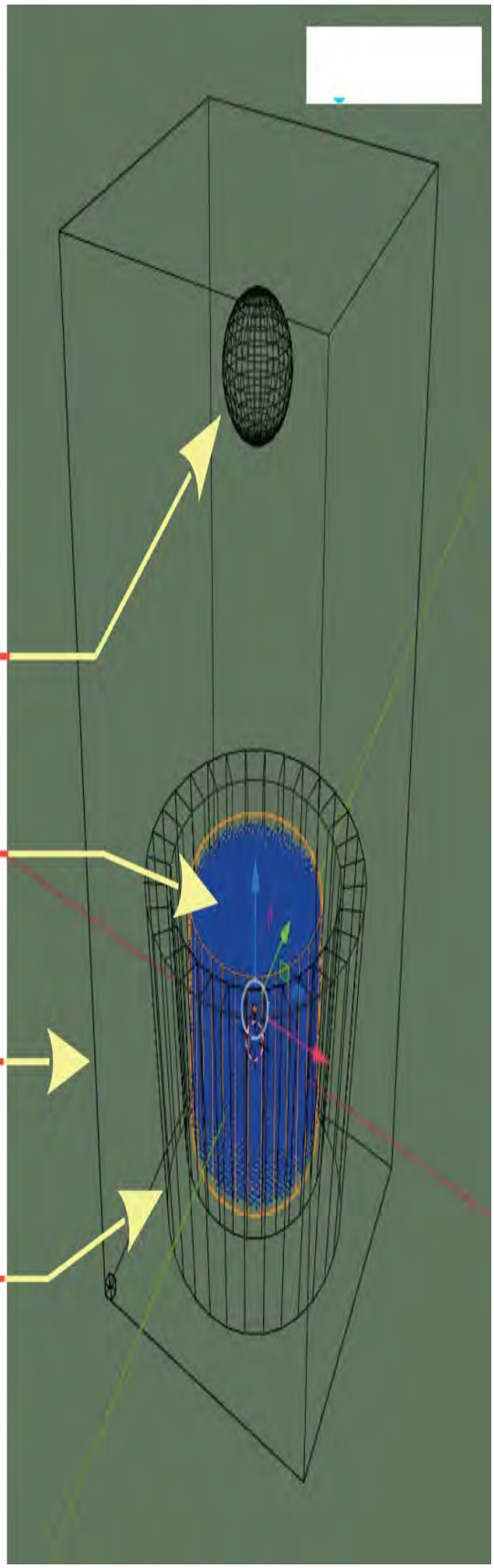


Figure 25.80

Arrangement.

The arrangement of Objects for this simulation is shown in Figure 25.81. A Cube has been scaled on the Z Axis forming the Domain. The Cup is scaled to fit inside the Domain and positioned at the bottom.

A Cylinder Object is Scaled to fit inside the Cup and positioned just above the bottom of the inner surface of the Cup. In Figure 25.81 Data has been Baked, therefore, Fluid Particles can be seen (blue dots) attached to the Cylinder while at Frame 1 in the animation.



UV Sphere – Fluid Type: Effector – Collision
Surface Thickness: 1.0000

Inflow Object Cylinder – Fluid Type: Flow
Flow Type: Liquid – Flow Behavior: Geometry

Domain Cube – Fluid Type: Domain
Domain Type Liquid

Cup – (Figure 25.80) Fluid Type Effector
Effector Type: Collision

Figure 25.81

A UV sphere Object is Scaled down and positioned towards the top of the Domain immediately above the Fluid Inflow Object.

Settings: In the Property Editor, Physics Properties, default settings are employed with the Objects participating in the Simulation as follows:

In the Domain settings the **Liquid** and **Mesh Tabs** are checked. The File path to the **Cache Folder** is entered in the Cache Tab. The End Frame in the **Cache Tab** and in the **Timeline End Frame** is **50** (Figure 25.82).

Note: In the **Cache Tab** the default **Type** is **Replay.** Check: **Is Resumable.**

Before Baking Fluid the **UV Sphere Effector Object** is animated to fall into the Cup starting at Frame 10 and at Frame 30.

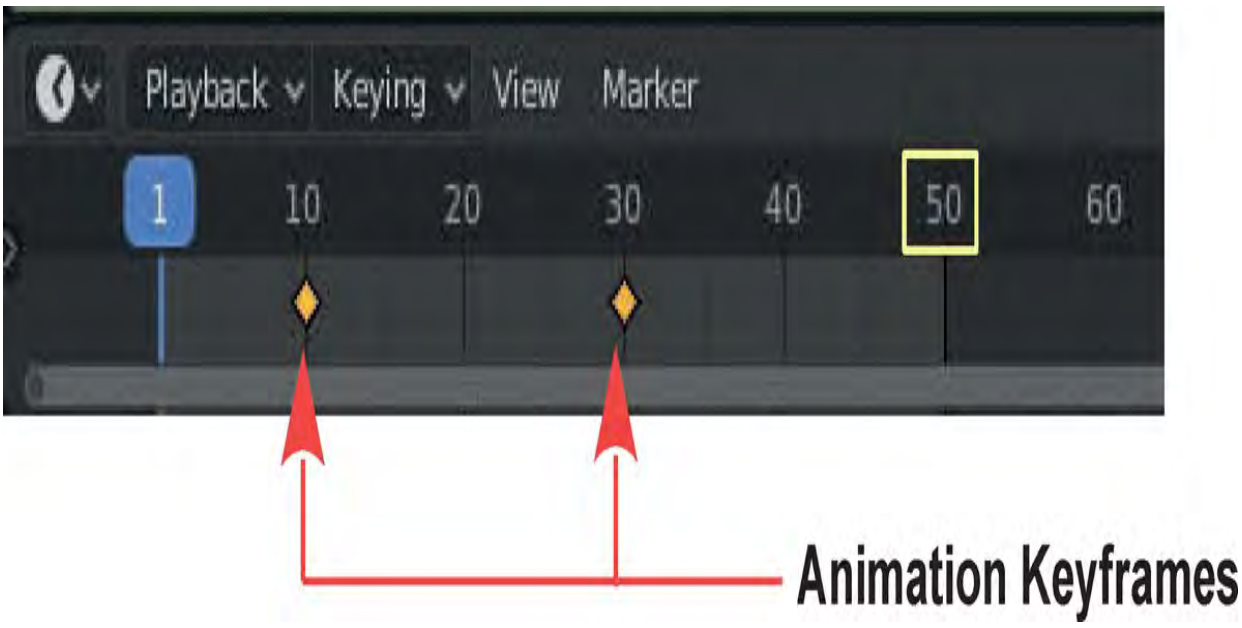
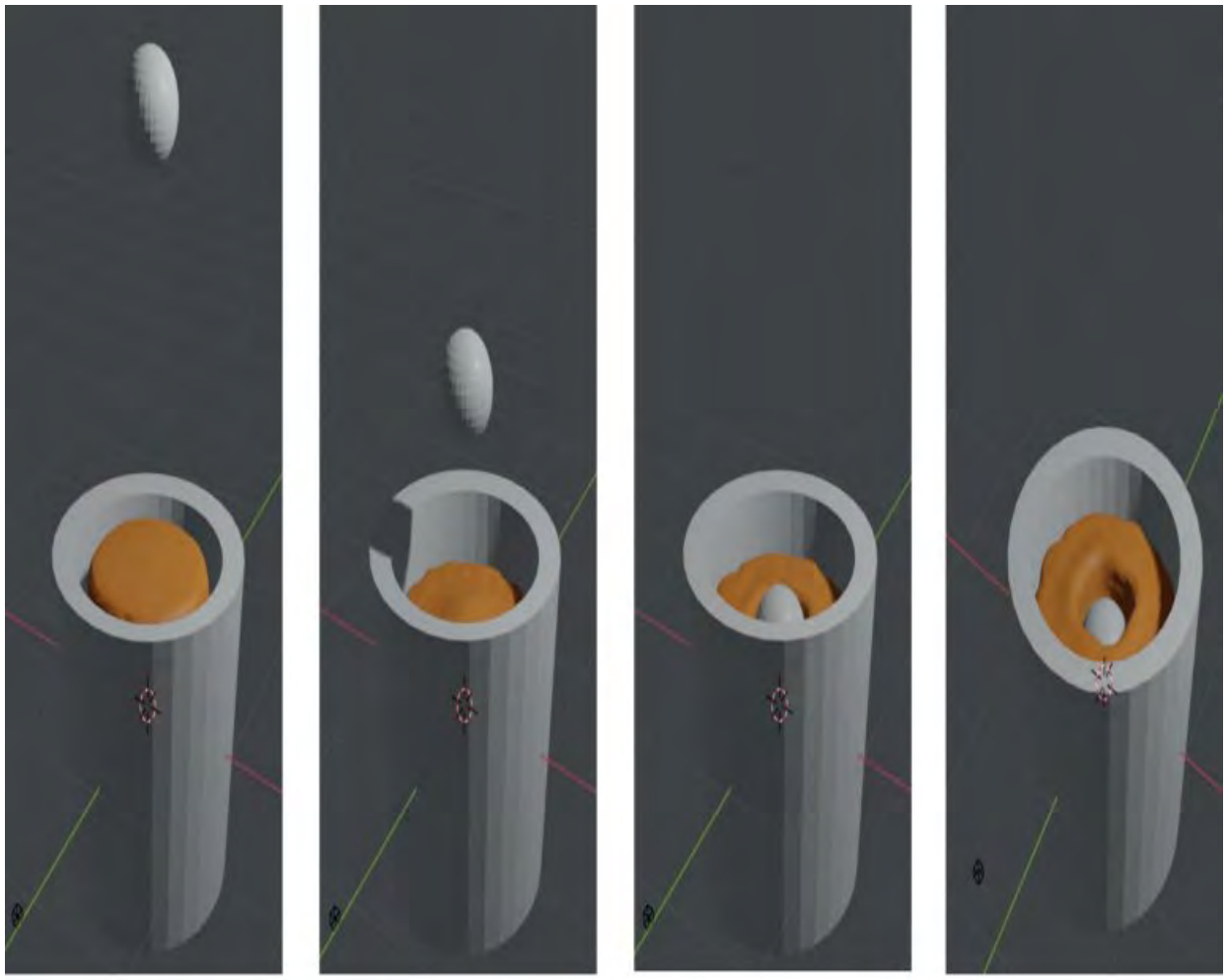


Figure 25.82

With the Domain selected, change the **Type** in the **Cache Tab** to **Modular** and ensure you check **Is Resumable.**

Bake Data and Mesh.

With Data and Mesh Baked, Rendered Viewport Shading Mode sees the Sphere splashing into the Liquid when the Animation is played.



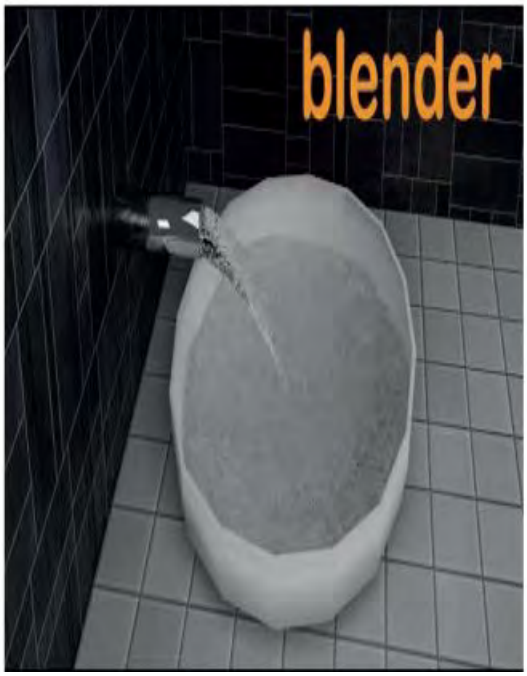

blender

Figure 25.83

25.11 Smoke and Fire Simulation

Simulating **Smoke and Fire** will crease fantastic effects. The setup is similar to Fluid Simulation with a few exceptions.

The forgoing being the case, revisit the **Quick Smoke Method** for **Fire** described in **Chapter 1 — 1.13.** By analysing the Properties Editor, Physics and Material Properties for this quick method you will gain an appreciation of the setup required.

The arrangement in the 3D Viewport Editor is simply two Cube Objects, one Scaled down and positioned inside the other. 25.85 shows the in VVireframe Display Mode.

Physics Properties

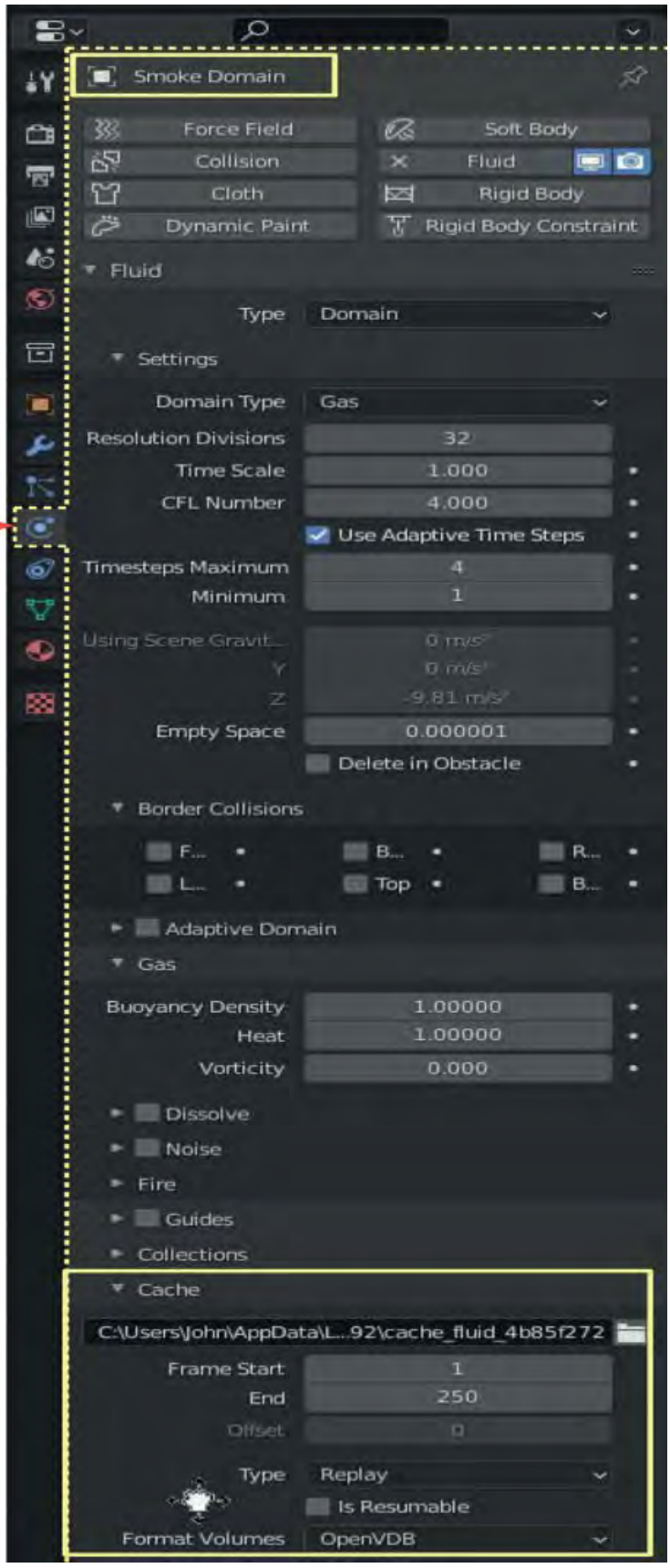


Figure 25.84

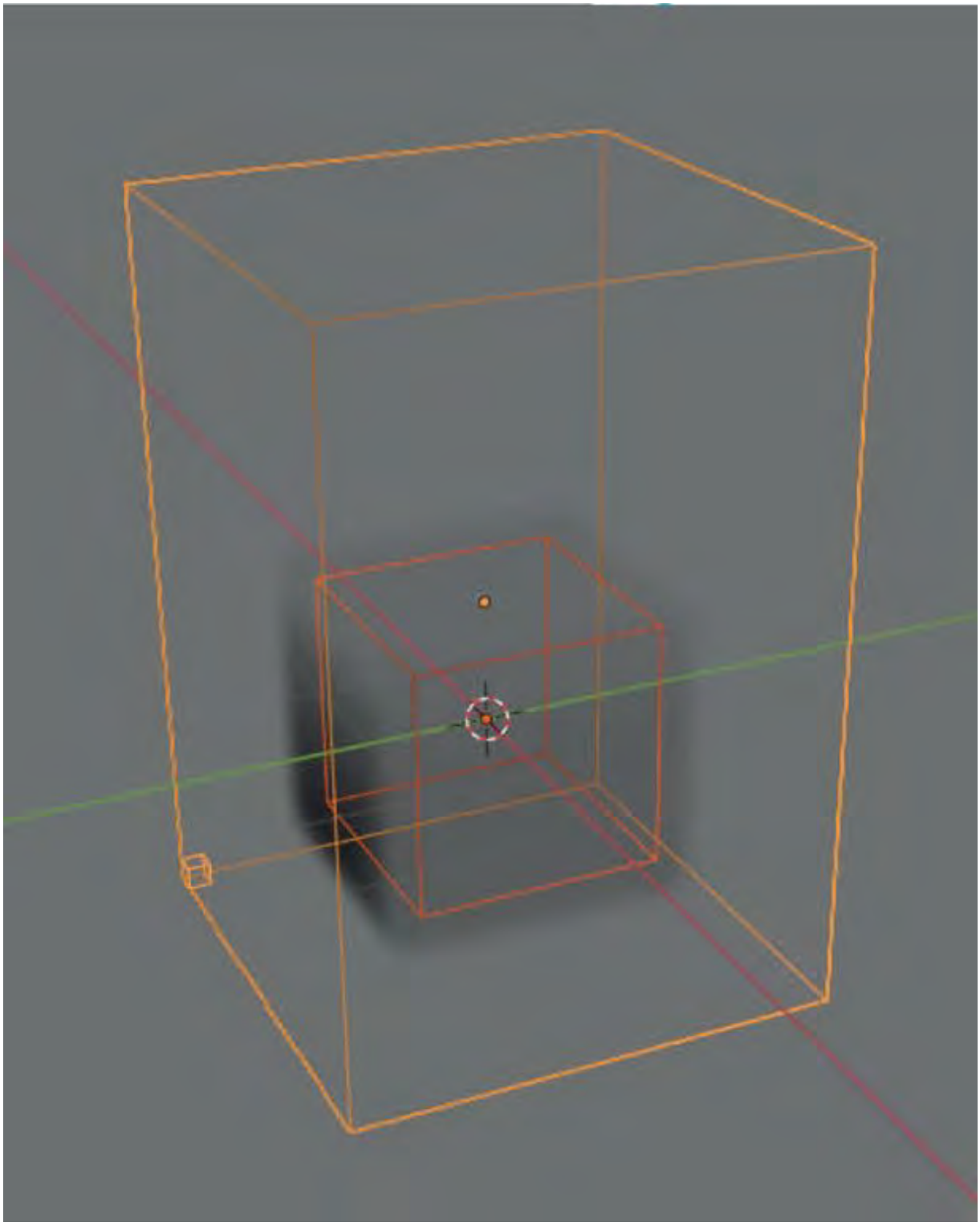


Figure 25.85

The **Properties Editor, Physics Properties** for the Domain are shown in Figure 25.84.

Note: When Quick Smoke is activated the File Path to the Cache Folder is set By Blender and the End Frame is 250 which is the same as the End Frame in the Animation Timeline.

When the Animation is played in the Timeline Editor you see Smoke issuing from the Cube Emitter and rising in the Domain.

Note: In the Cache Tab the Type is **Replay** and **Is Resumable** is unchecked.

25.12 Smoke and Fire from Scratch

The Quick Smoke simulation was used as an introduction in Chapter **1** — 1.13. This **Quick Method** (section 25.11) of producing the simulation is a good place to begin but when you understand the procedure for generating Smoke and Fire from first principles you will have much more control and be able to use the full potential of the tool. This section will not attempt to cover the topic in its entirety, instead it will give you a basis which should make following tutorials easier.

Suggested Pre-Requisites

3D View Editor Background

In Chapter **1** the diagrams demonstrating the affect showed the default Cube Object burning against a pale green **3D Viewport Editor background.** If you are starting from scratch the default Screen background is dark gray and is not the best when viewing gray smoke. It is, therefore, suggested you change the the color of the background (see Chapter 2 — 2.18).

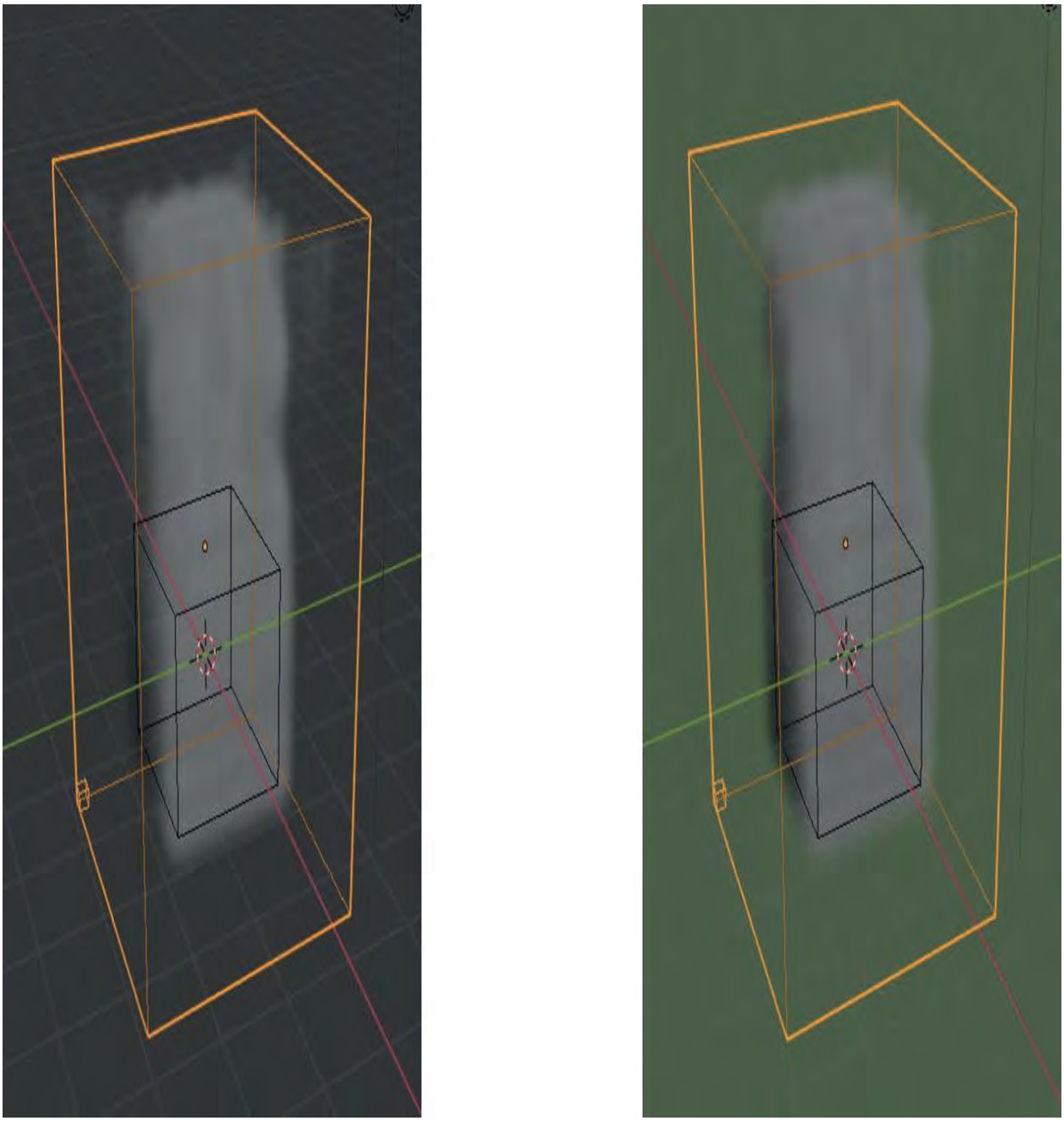


Figure 25.86

World Background

Figure 23.76 shows the 3D Viewport Editor background in Solid Viewport Shading Mode (see Chapter 14). **If** you switch to Rendered Viewport Shading Mode you will be back to a gray background color. Rendered Viewport

Shading is previewing what you see when the Scene is Rendered. You change this background in the World Settings (see Chapter 14 — 14.9).

Lighting

The default Blender Scene contains a single Point Light (see Chapter 15). Smoke, being what it is, can be very thin and wispy in nature which makes it difficult to see at times. It is suggested that you, at least, change the Point Light to an **Area Light** in the **Properties Editor, Object Data Properties** and ramp up the **Power setting to 3000w.** You could also consider duplicating the lamp a number of times and positioning the duplications around the Scene.

Render Settings — Volumetrics

Render Properties

Figure 25.87

When dealing with Smoke, **Volumetric Lighting** is desirable to attain the effect of volumetric scatter 16 —16.22).

Volumetrics are found in the Properties Editor, Render Properties (Figure 25.87). For the Smoke demonstration check

The default settings will be used with one exception. Change the **Tile Size** to **2px.** The lower the Tile Size value the more detail in the render.

Withe the preliminary set up parameters in place you are good to go with Smoke and Fire simulation.

Recap on the Quick Smoke Method

To start with, take a look at the Smoke Quick Method and the three options, Smoke, Fire and Smoke + Fire.

The three images in Figure 25.88 show the Quick Smoke Options using the default Cube Object. To activate Quick Smoke, select the Cube, then in the 3D Viewport Editor Header, click Object, go down in the menu that displays and select Quick Method, Smoke. The Cube is displayed in Wireframe surrounded by a larger Cuboid. The larger Cuboid is the Domain which is a cubic volume of 3D Space in which the simulation takes place. The smaller Cube is the Emitter, the emitting Object.

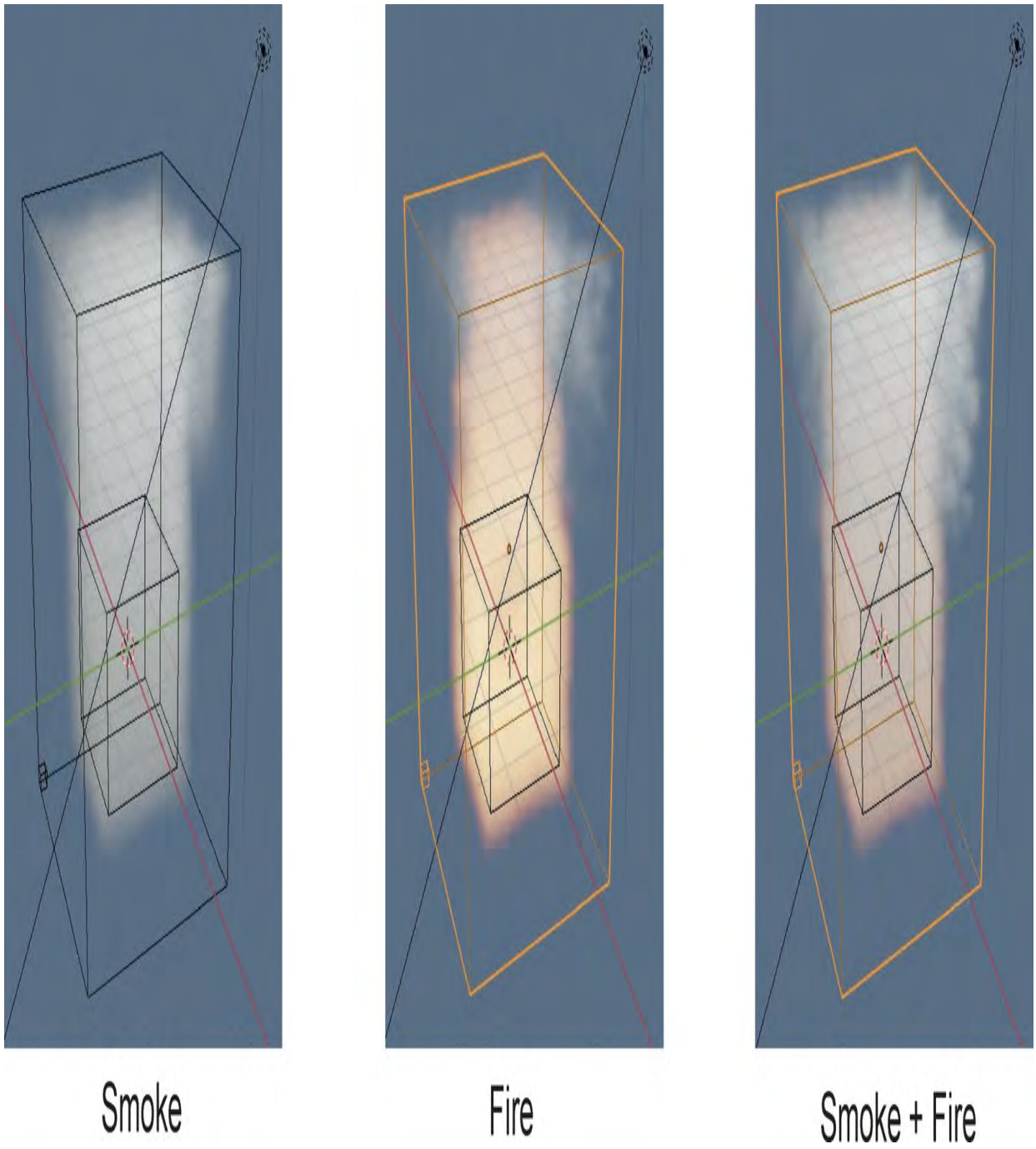


Figure 25.88

When Quick Smoke is activated a panel displays in the lower LH corner of the 3D Viewport Editor where you click and select the options.

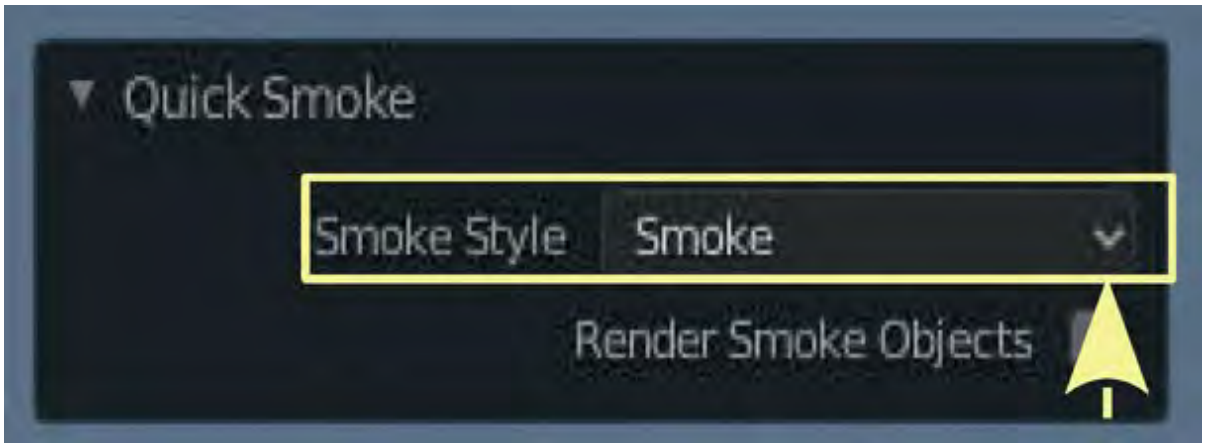


Figure 25.89

Note: If you perform an operation in the 3D Viewport Editor the Quick Smoke options panel disappears.
Losing the Quick Smoke options emphasises the need for knowing what settings the Quick Smoke method has automatically generated.

When you select an object in the 3D Viewport Editor and apply the Quick Smoke tool you are designating the Object as a smoke **Emitter.** Blender automatically applies Physics settings to the Object. Blender also creates a cubic volume of 3D Space in which the simulation will take place ; the **Domain.** Physics settings are also applied.

Smoke from Scratch

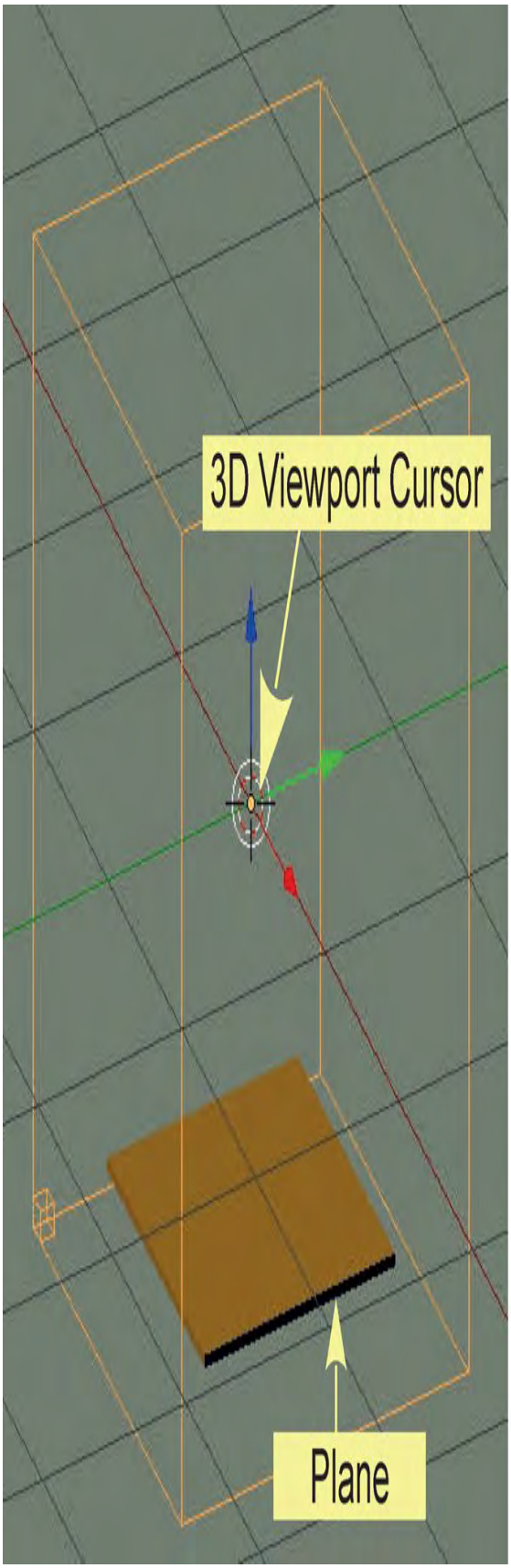
The Quick Smoke method is an ideal starting point for Smoke Simulation but you should understand the basic procedure of setting up a simulation from first principles to make it easier in understanding tutorials. You will also have greater control of the Smoke effect. Smoke simulation will produce fantastic effects but it does entail some fairly complex procedures. This section will not attempt to cover the topic in its entirety, instead it will give you a basis which should make following tutorials easier.

Smoke can be generated using any Mesh Object in a scene as an **Emitter (Flow Object).**

There are basically only two objects required in a scene for a smoke simulation, a **Domain** and an **Emitter.** This is similar to the principles for setting up **Fluid Simulation** (see 23.7). Both the Domain and the Emitter will have Physics applied.

Scene Setup

Set the Scene as shown in Figure 25.90. A Plane Object will be used as the **Emitter** (source of the smoke) and a Cube Object will be used as a **Domain.** Scale the Cube and the Plane as shown making sure the **Plane** is totally inside the **Domain** (Check in **Front Orthographic view).** For the demonstration keep the Objects in proportion with the Domain Cube approximately 1.5 Blender units high.


3D Viewport Cursor
Plane

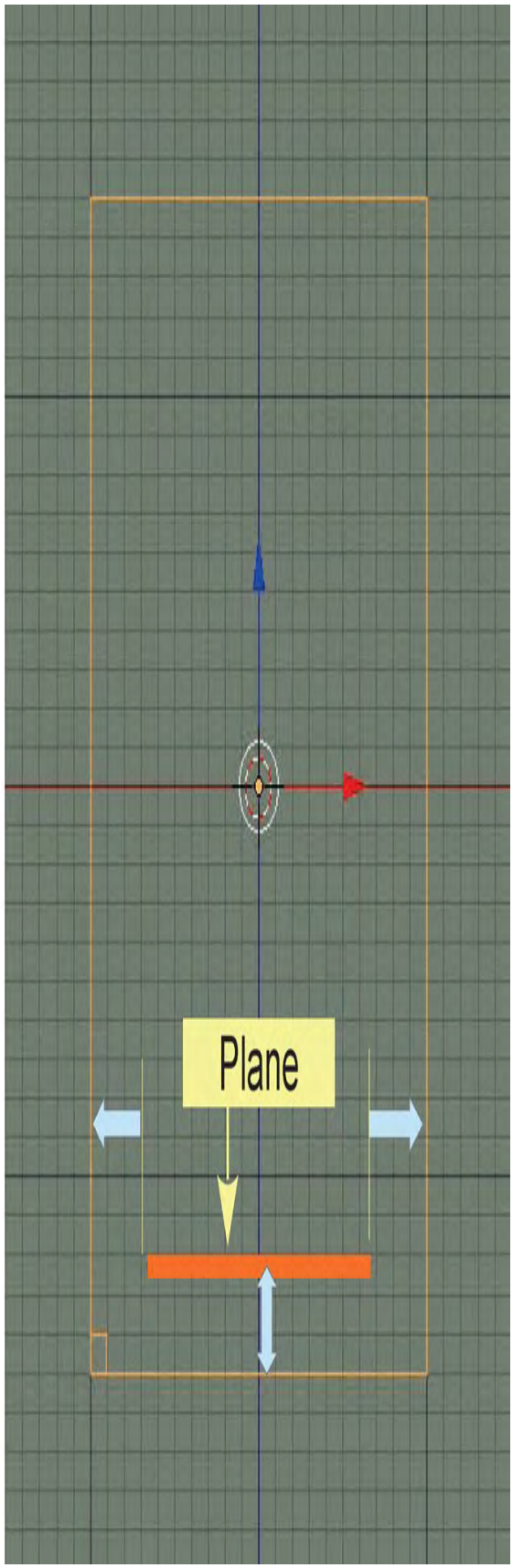

Plane

Figure 25.90

You can use Wireframe Viewport Shading to position the Objects then change to Solid Viewport Shading. In Solid Mode the Plane is inside the Cube, therefore, you only see the Cube.

To use the Objects in a Smoke simulation **Physics** has to be applied.

The Domain Object (Cube)

To emit smoke from a Mesh Object you have to define a volume of space in which the smoke simulation takes place. In Blender Physics this volumetric space is the **Domain. Note:** If you use the default Cube in the 3D Viewport, this Cube has a default **Material** applied. A new Cube Object introduced to the Scene will not have Material.

With the Cube selected go to the **Properties Editor, Physics Properties** and enable **Physics** for **Fluid . Set Fluid Type:Domain, Settings Tab, Domain Type Gas** (Figure 25.91).

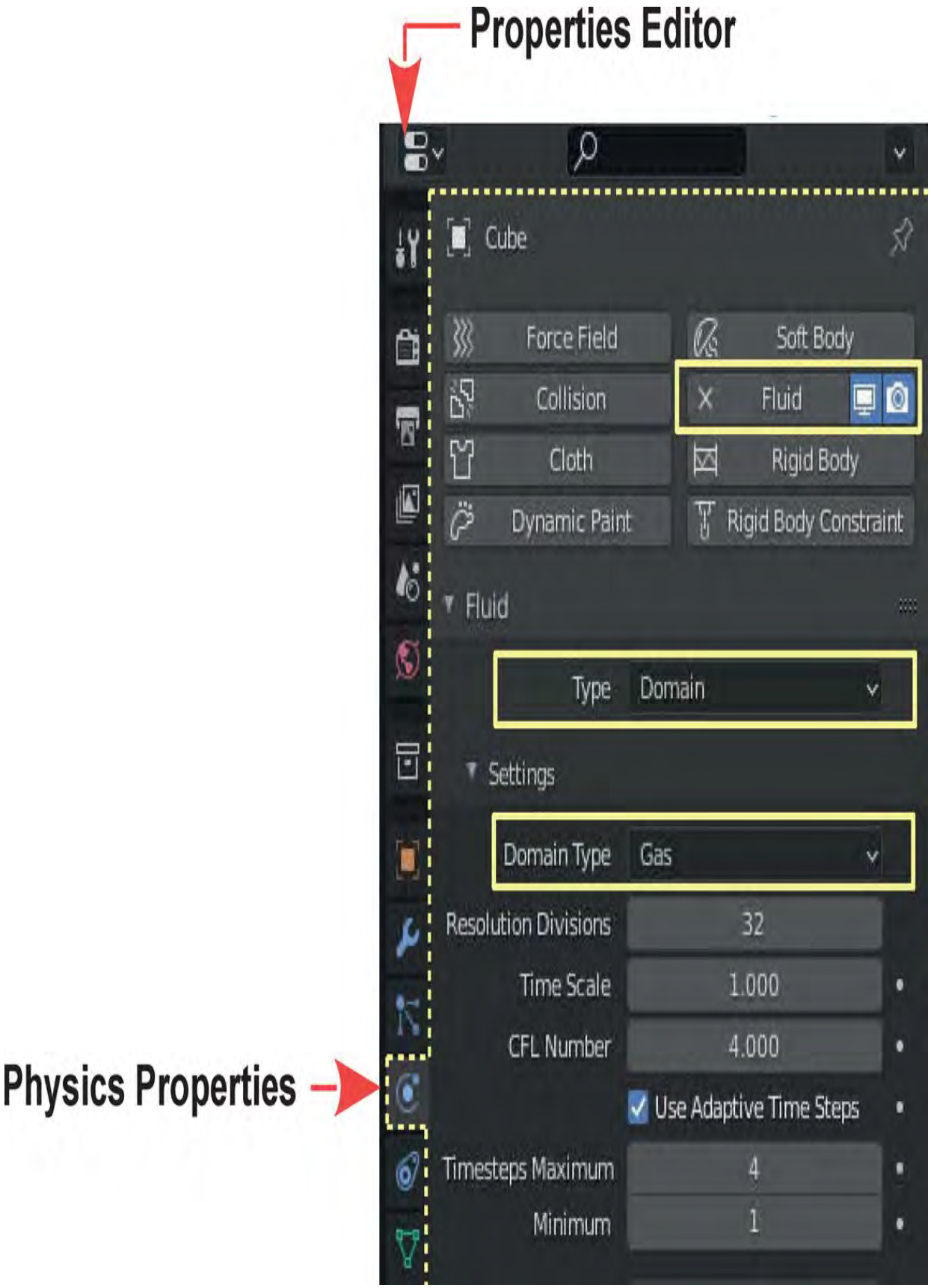

Properties Editor
Cube
Force Field
Soft Body
Collision
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Fluid
Type
Domain
Settings
Domain Type
Gas
Resolution Divisions
32
Time Scale
1.000
CFL Number
4.000
Use Adaptive Time Steps
Timesteps Maximum
4
Minimum
1
Physics Properties

Figure 25.91

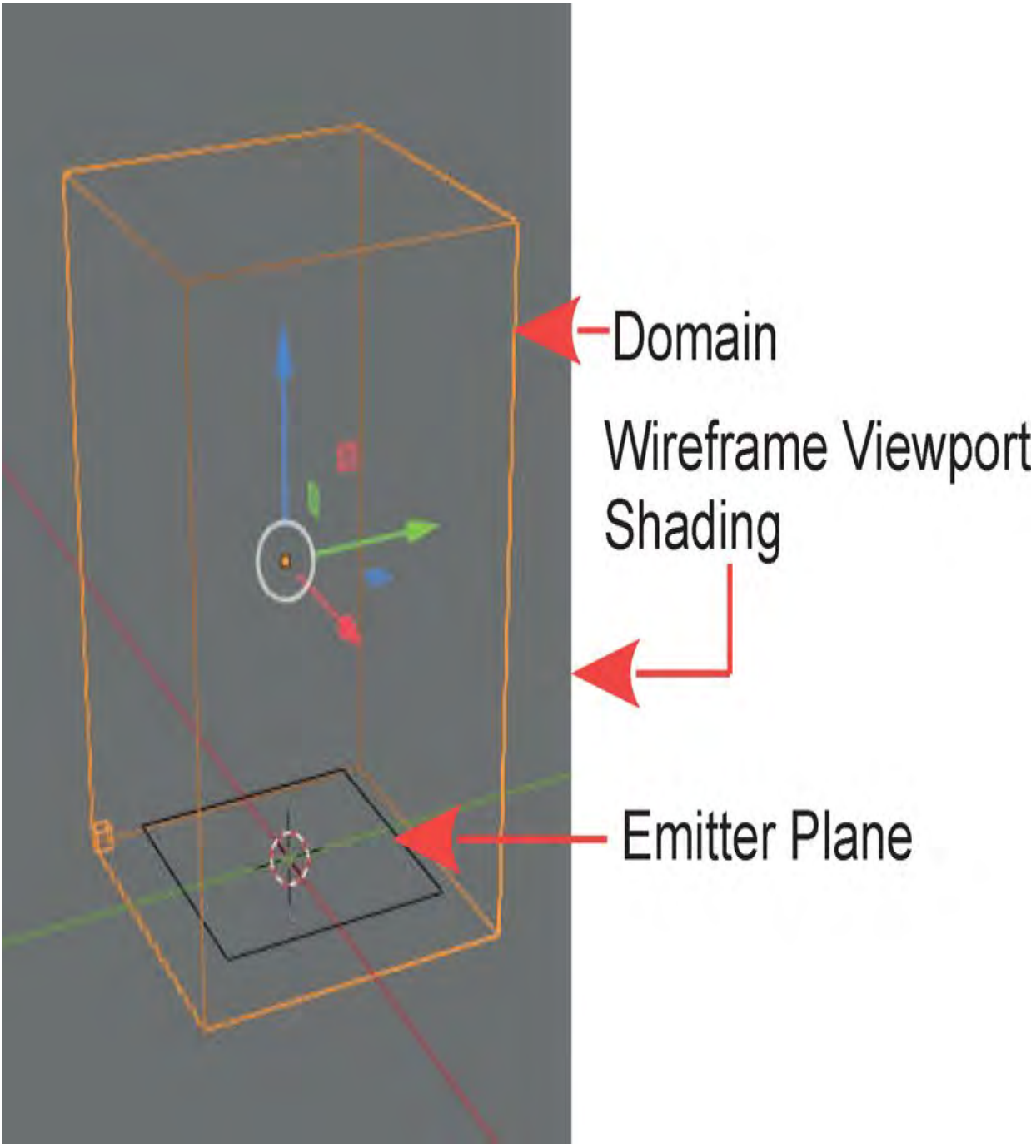


Figure 25.92

The Emitter Object (Plane)

Select the **Emitter Plane** and set **Physics Properties** as shown in Figure 25.93.

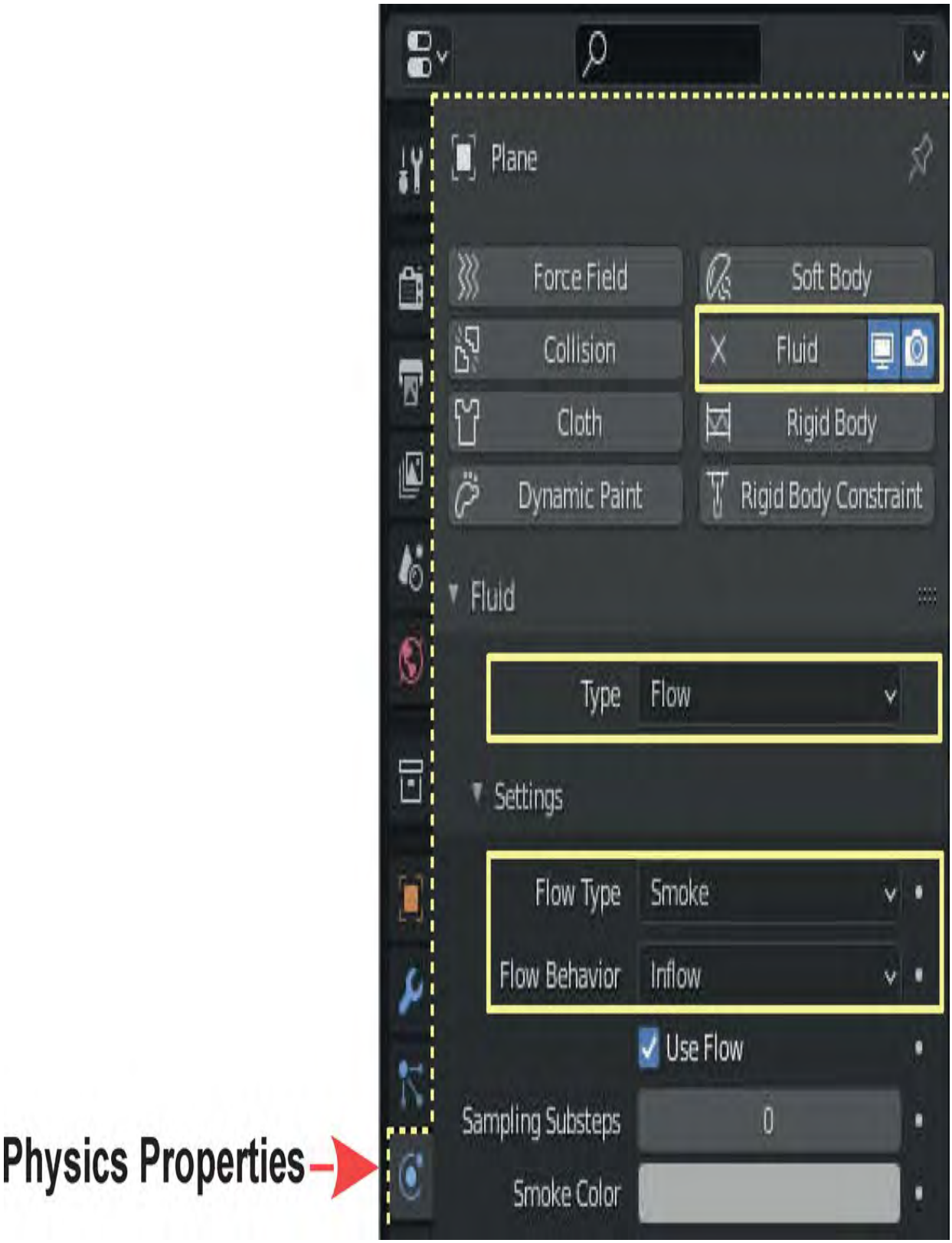

Plane
Force Field
Soft Body
Collision
Fluid
Cloth
Rigid Body
Dynamic Paint
Rigid Body Constraint
Fluid
Type
Flow
Settings
Flow Type
Smoke
Flow Behavior
Inflow
Use Flow
Sampling Substeps
0
Smoke Color
Physics Properties

Figure 25.93

Play the Animation

At this stage playing the Animation in the Timeline Editor shows Smoke being Emitted from the Plane and rising in the Domain.

Note: In the Properties Editor, Physics Properties for the Domain, Blender has assigned a Cache Folder. The **Cache Mode is: Replay** and **Is Resumable** is NOT checked.

Note: The default **End Frame** values are being used (250 Frames).

Smoke will issue from the Plane and rise in a column towards the top of the **Domain** Cube (Figure 25.94). This simulation will play for the default 250 Frames as set in the Timeline Editor and then repeat. Press **Esc** to quit or press the **Pause** button in the Timeline Editor and then the Return to Start button.

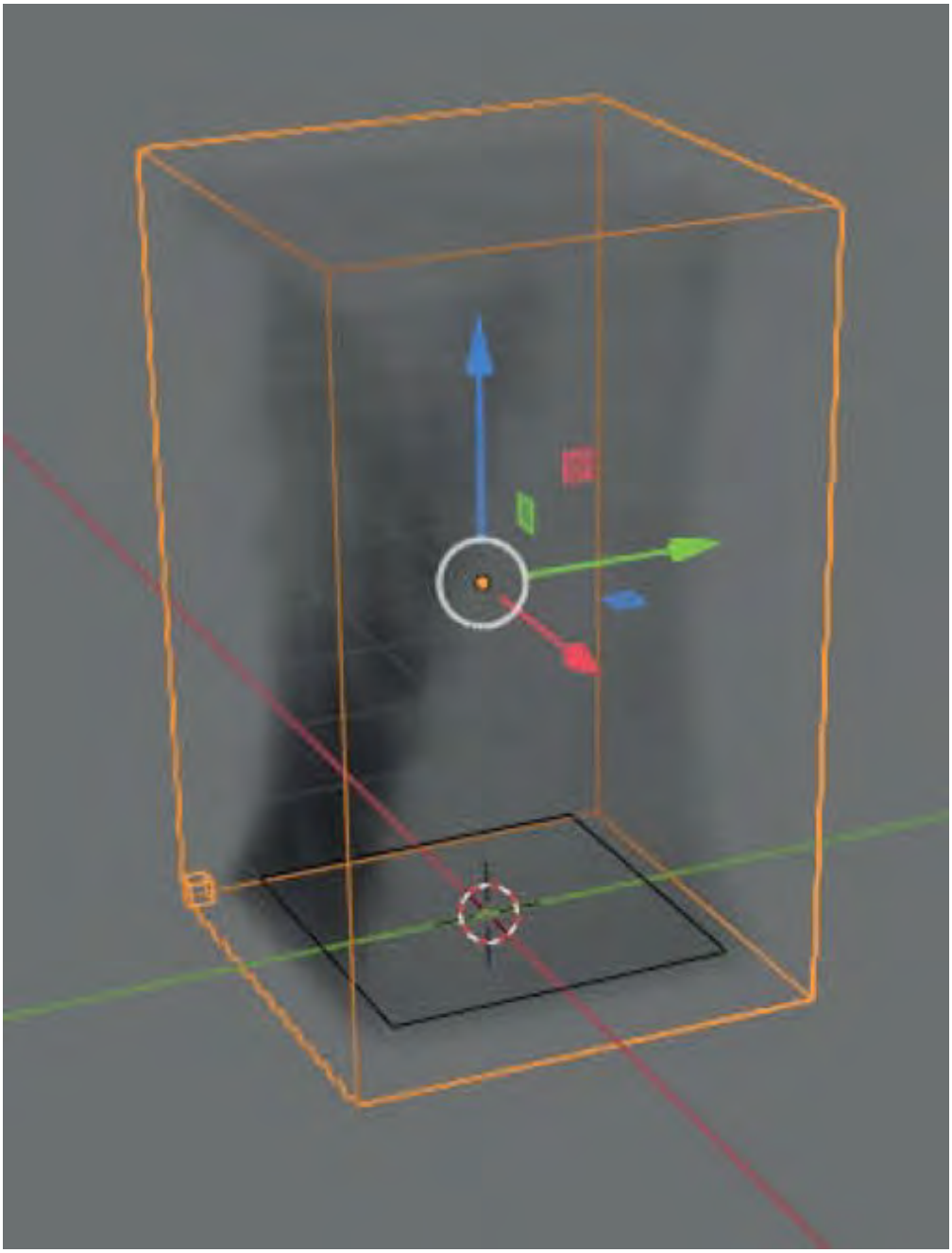


Figure 25.94

Figure 25.95

Note: At this point the 3D Viewport Editor is in **Wireframe Viewport Display Mode** and with the animation paused in the Timeline you will see smoke as shown in Figure 25.94. If you switch to Rendered Viewport Shading Mode you will be disappointment to find that the Smoke doesn't show. If you press F12 to render the Scene all you see is the domain as a solid Object.

To display Smoke in Rendered Viewport Shading Mode and in a Render, Nodes have to be configured in the **Shader Editor.**

Viewport Display

How a Smoke simulation displays in the Viewport is dependent on the Material Nodes for the Domain Object. If you have created the Domain using a new Cube Object, the Cube will not have a material applied, therefore, the Material buttons in the Properties Editor will only contain the **New** button and the Shader Editor will be empty (NO Nodes).

When using a New Cube Object click the New button to display Material controls. If you have used the default Cube for the Domain, Material controls will be displayed and a default Node arrangement will be in the Shader Editor (Figure 25.96).

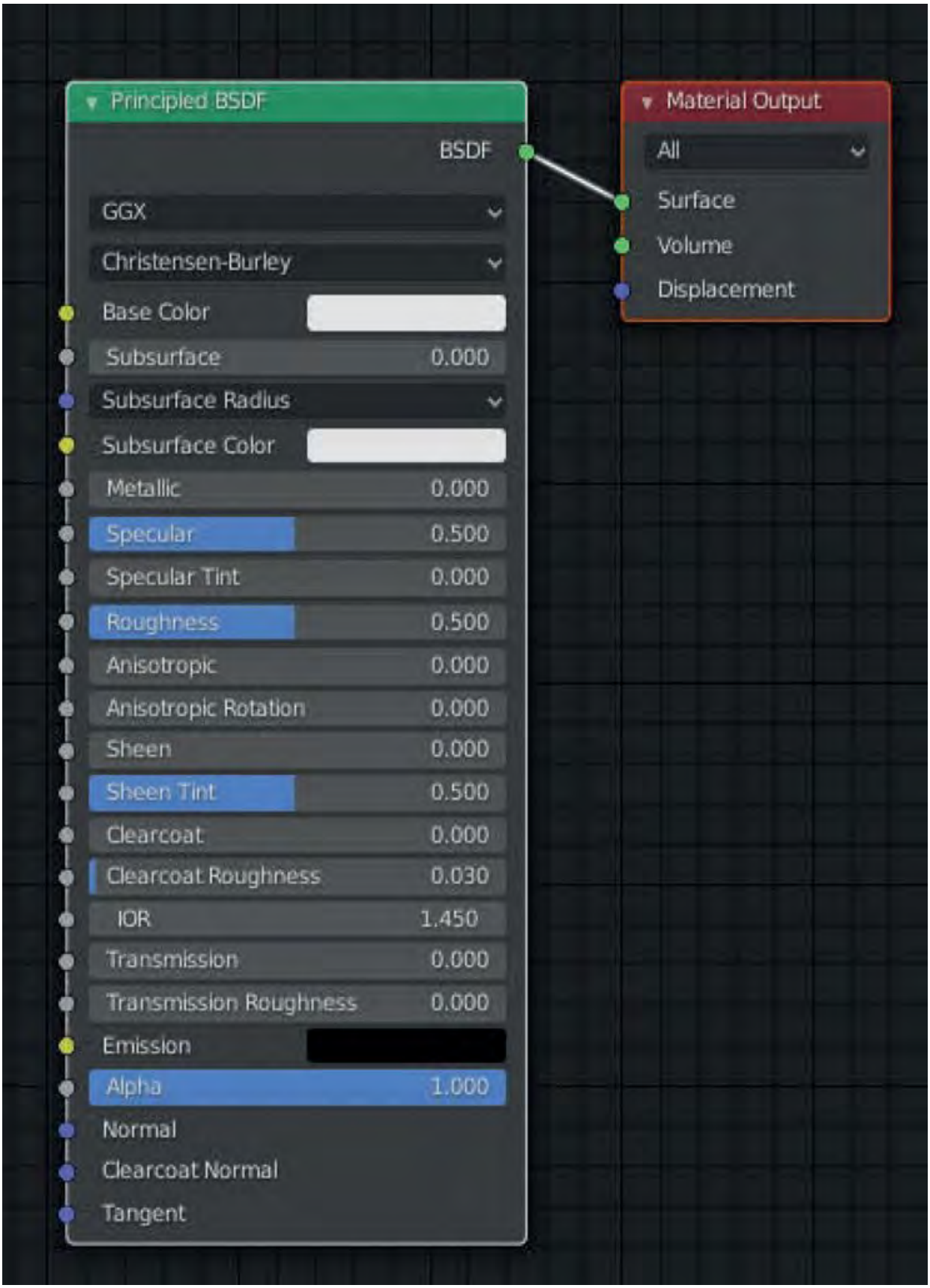

Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius
Subsurface Color
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.500
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement

Figure 25.96

The Node arrangement consists of a Principled BSDF Node connected to a Material Output Node. Smoke does **NOT** show in Rendered Viewport Shading Mode.

Replace the Principled BSDF Node with a **Volume** Node and connect as shown (Figure 25.97).

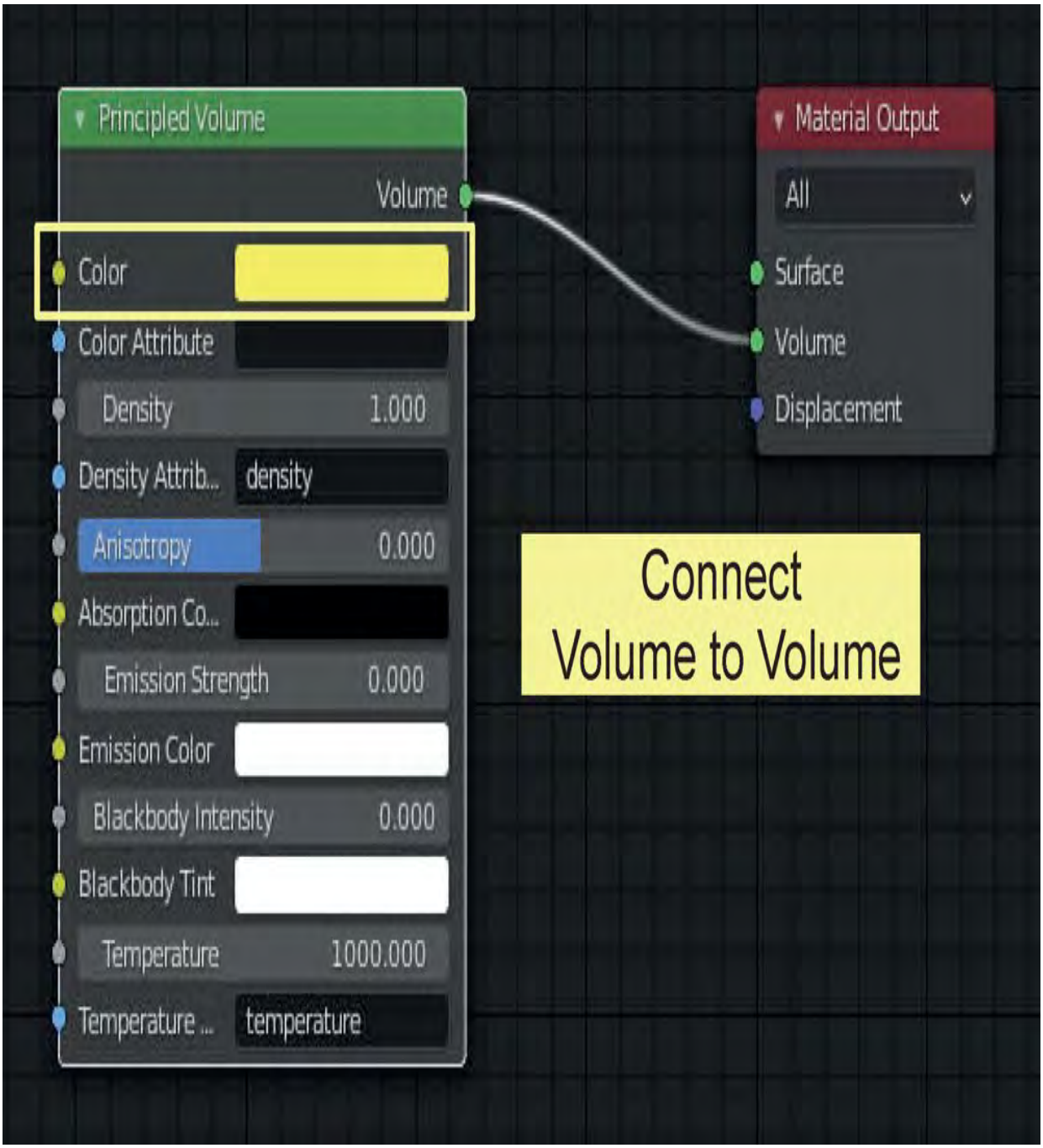


Figure 25.97

With the Node arrangement reconfigured Smoke will display in Rendered Viewport Shading Mode and in a Render when you press F12.

Note: The Render will take a few seconds to load.

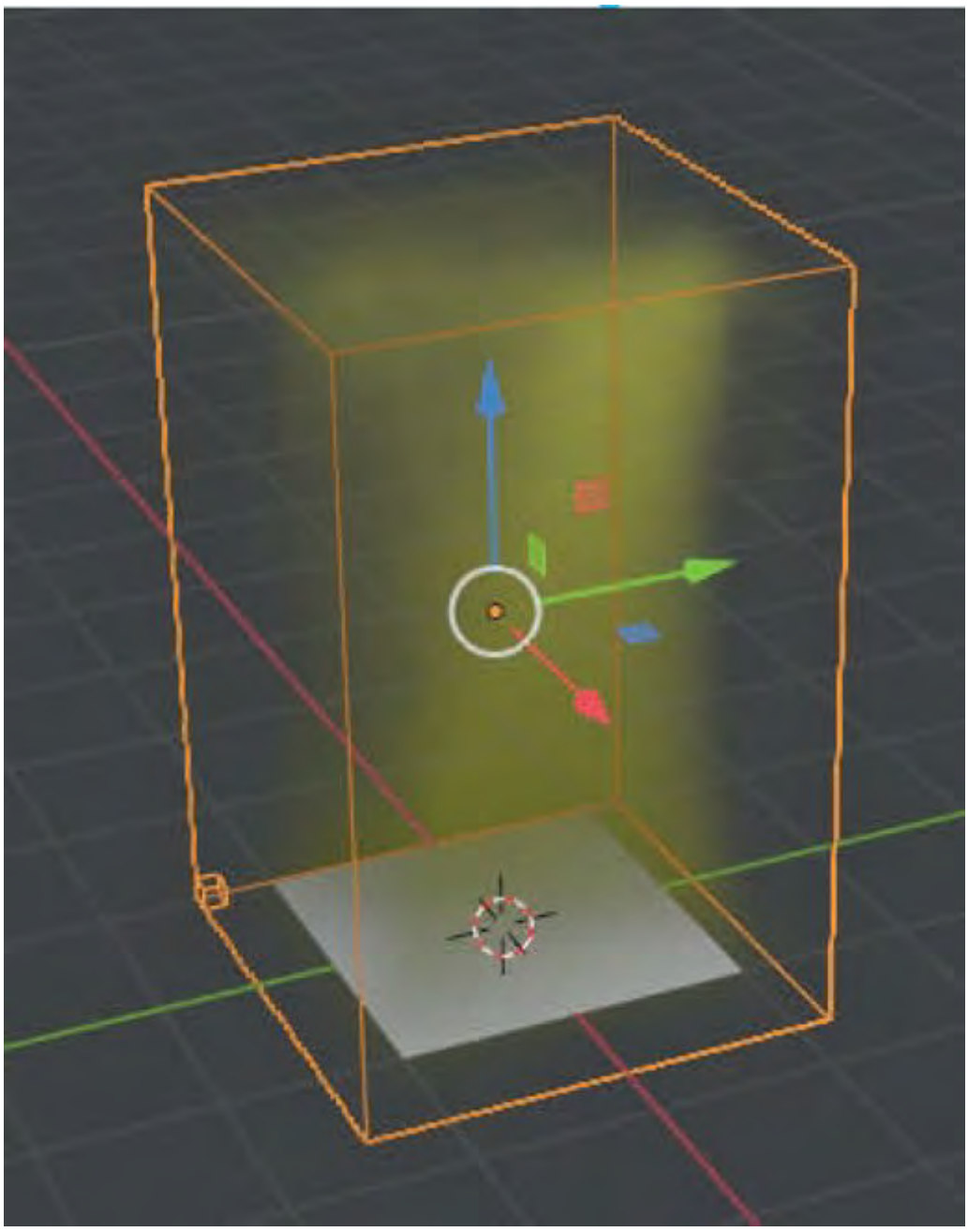


Figure 25.98

At this point you can control the **Smoke Color** in the Principled Volume Node **with the Viewport in Rendered Viewport Shading Mode.**

Note: There are other Smoke controls in the Physics buttons for the Emitter Plane. The Smoke Color control in these controls will allow you to change the Smoke color when in Solid Viewport Shading Mode.

At this stage only Smoke Simulation has been considered. Its time to look at **Fire** which is a little different.

Fire Simulation

Emitter Physics

Using the previous Object, Node and settings for the Emitter and the Domain make one change in the Emitter Plan Physics. In the **Settings Tab** change the **Flow Type to Fire.** Play the animation in the Timeline while the Viewport is in **Solid Viewport Shading Mode** to see Fire instead of Smoke.

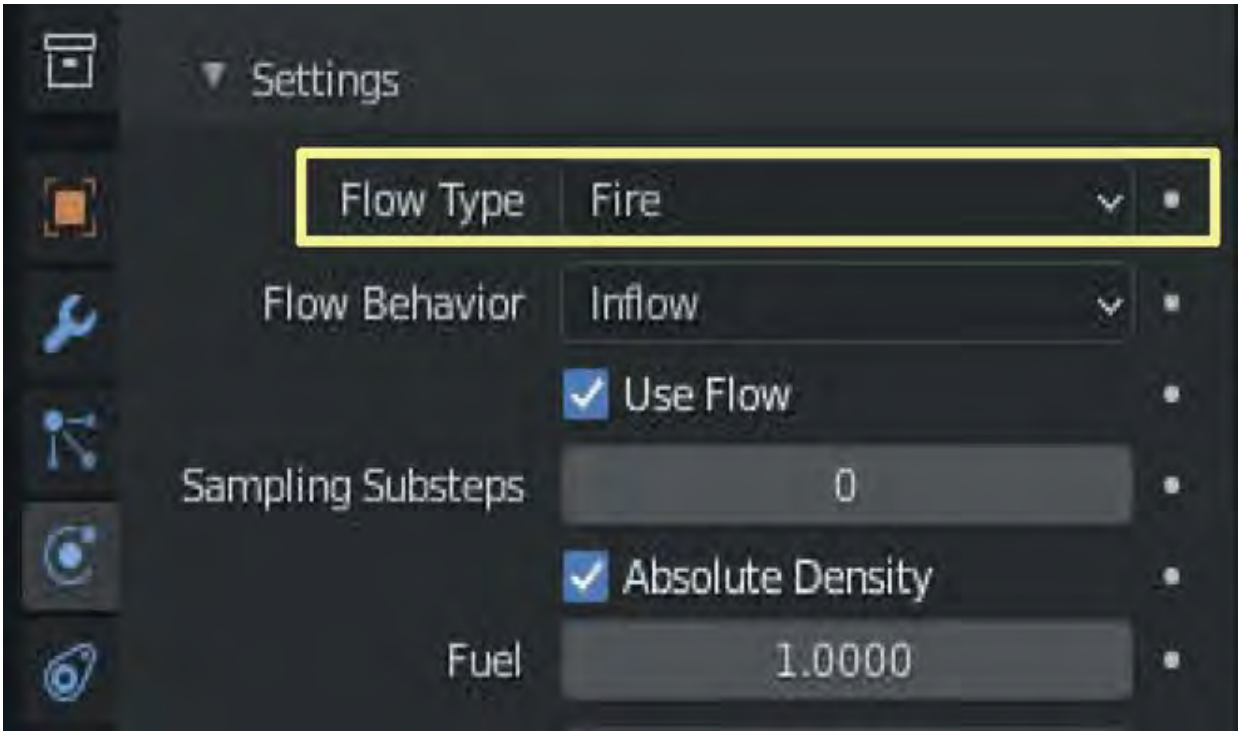


Figure 25.96

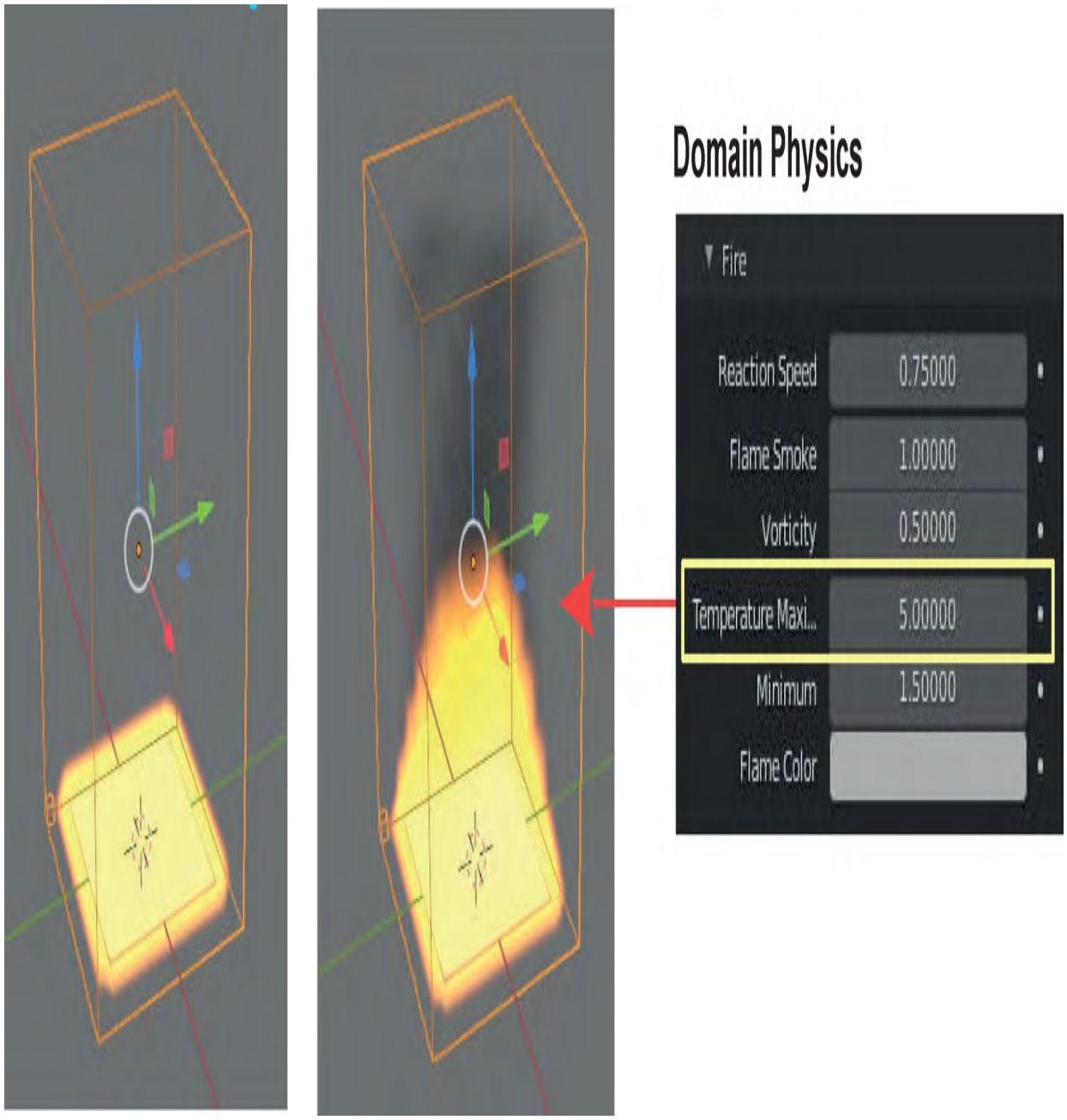


Figure 25.100 Domain Physics
Figure 25.101

Increase the **Temperature: Max.** in the Domain Physics settings, **Fire Tab.**

You guessed it! When you change to Rendered Viewport Shading Mode there is no Fire, maybe just a little bit of Smoke.

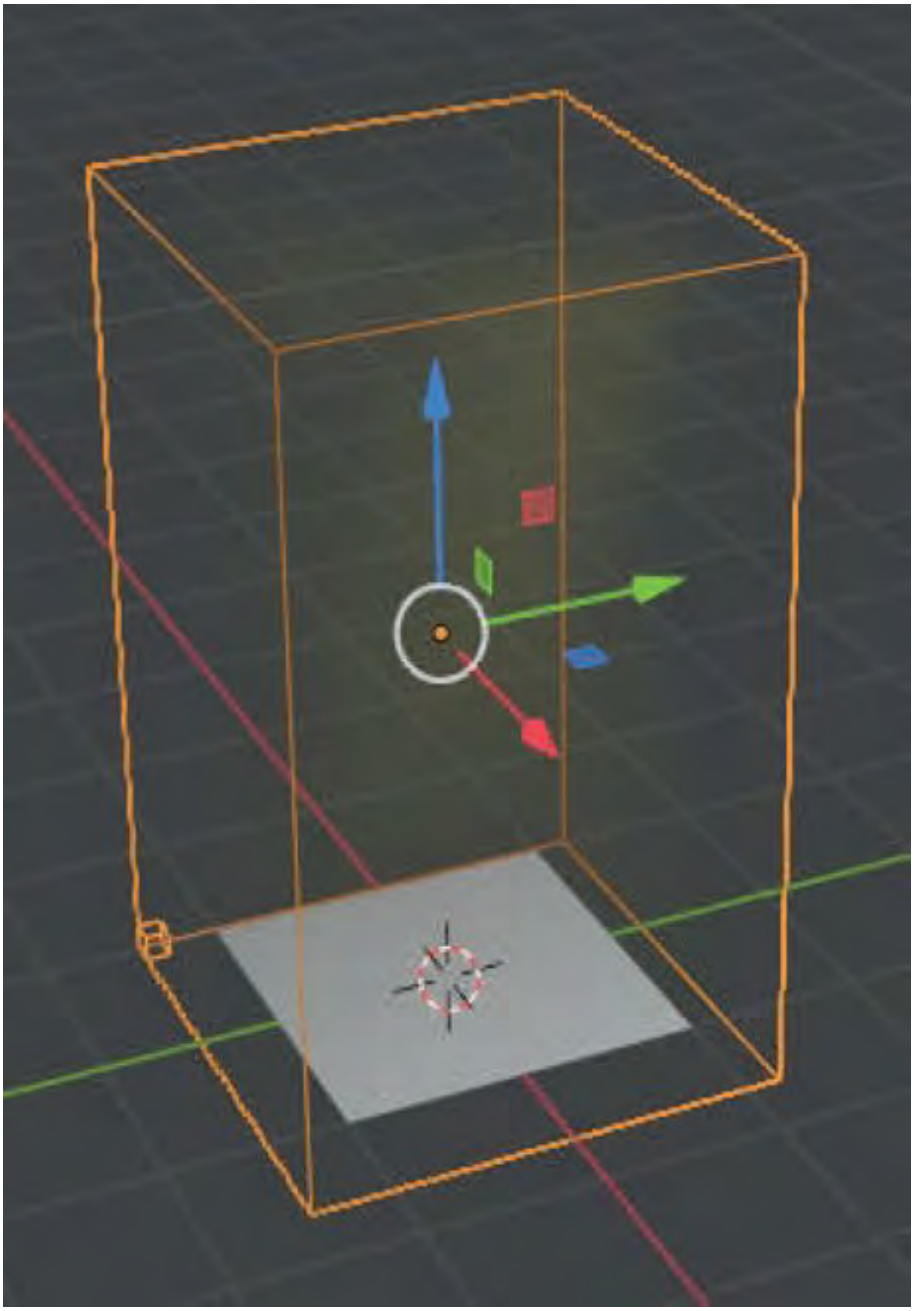


Figure 25.102

With the Domain selected the Shader Editor has the Principled Volume Node connected to the Material Output Node but with Fire this doesn't do the trick.

Further Node configuration is required. With the Domain selected add and connect Nodes as shown in Figure 25.103.

Add Category Input: Attribute
Multiply: Converter Math (change Add to Multiply)
Shader: Emission Shader-Add Shader

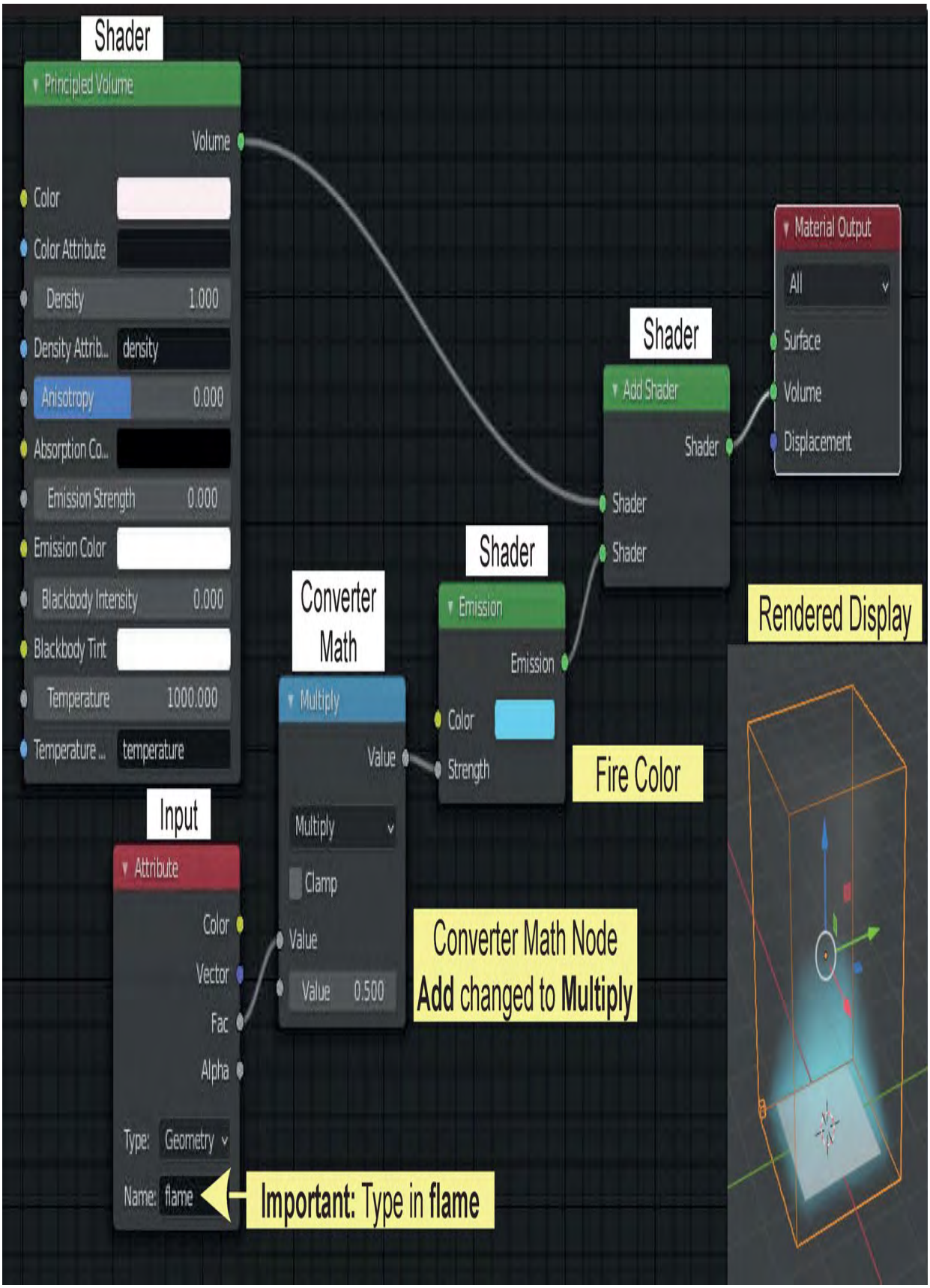

Shader
Principled Volume
Volume
Color
Color Attribute
Density 1.000
Density Attrib... density
Anisotropy 0.000
Absorption Co...
Emission Strength 0.000
Emission Color
Blackbody Intensity 0.000
Blackbody Tint
Temperature 1000.000
Temperature ... temperature
Material Output
All
Surface
Volume
Displacement
Shader
Add Shader
Shader
Shader
Shader
Emission
Emission
Color
Strength
Converter Math
Multiply
Value
Multiply
Clamp
Value
Value 0.500
Fire Color
Rendered Display
Input
Attribute
Color
Vector
Fac
Alpha
Type: Geometry
Name: flame
Converter Math Node
Add changed to Multiply
Important: Type in flame

Figure 25.103

Note: The Fire displays in different ways depending on the Viewport Shading Mode. The inset in Figure in Figure 25.103 is in Material Preview and Rendered Viewport Shading, Figure 25.104 is in Wireframe and Solid Viewport Shading.

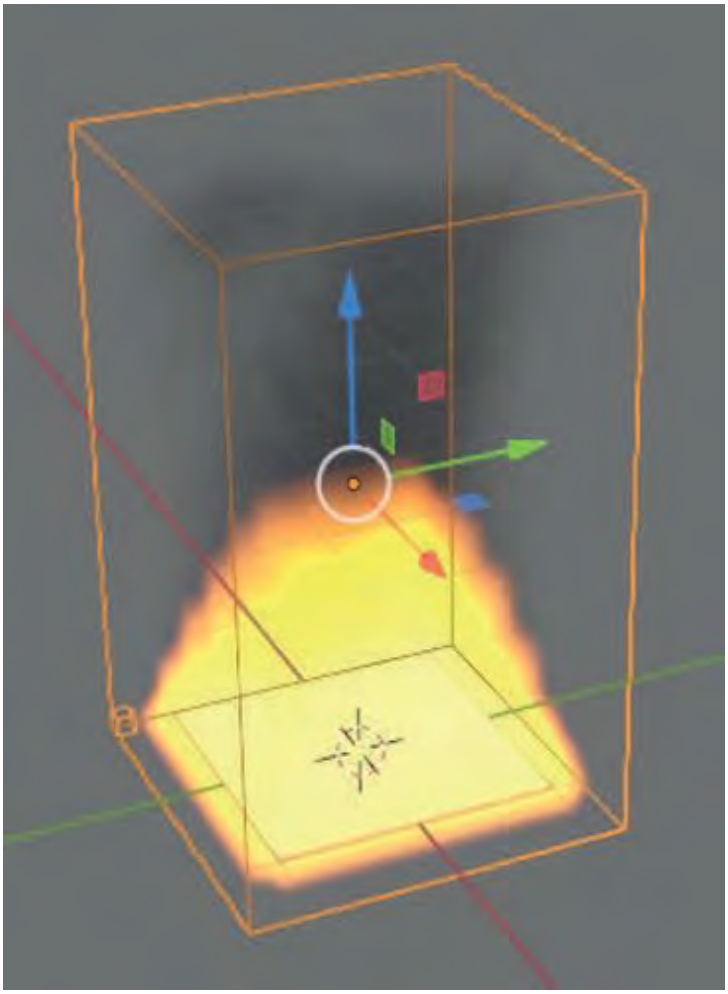


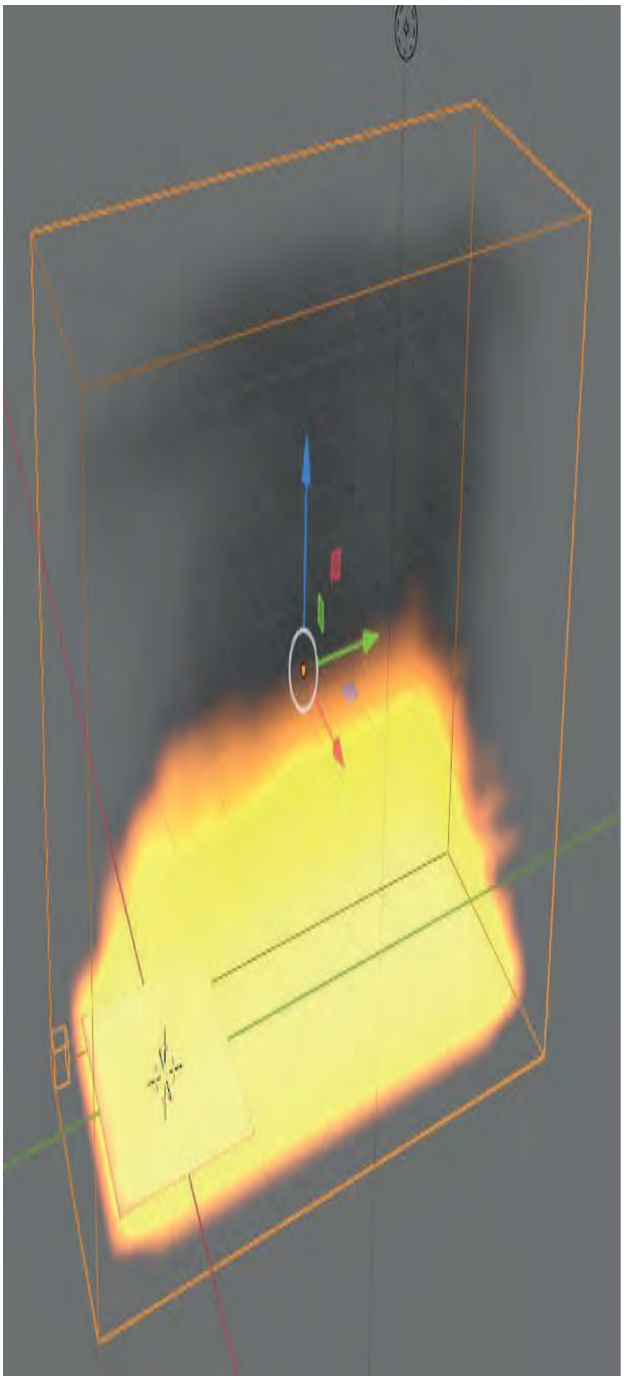
Figure 25.104 Solid Viewport Shading

Fire Simulation

With the identical Node arrangement in Figure 25.103 you may select the Emitter Plane and change the Settings, **Flow Type** to **Fire + Smoke.** When the Animation is replayed the Smoke will intensify and fill the Domain.

Domain Bounds

How the Fire and Smoke accumulates in the Domain is determined by the **Domain Bounds.** Every Object in Blender has a Bounding Volume which is a cubic volume of space in the Scene which it occupies for the purpose of calculation. By Scaling the Domain in the demonstration, along the Y Axis and repositioning such that the Emitter Plane is to one side you will see the Smoke and Fire being generated as shown in Figure 25.105.



Note: The Fire and Smoke filling the Domain is determined by the **Bounding Volume** of the Domain.

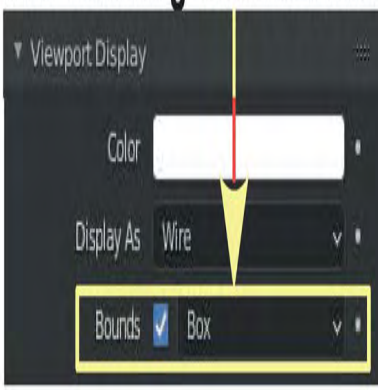


Figure 25.105

Domain Scaled on the Y Axis. The Bounding Volume is the same as the Scaled Cube.

When the Domain is reshaped in Edit Mode and the Animation played you see that the new Domain shape has no affect on the Smoke and Fire. The

display is governed by the Bounding Volume.

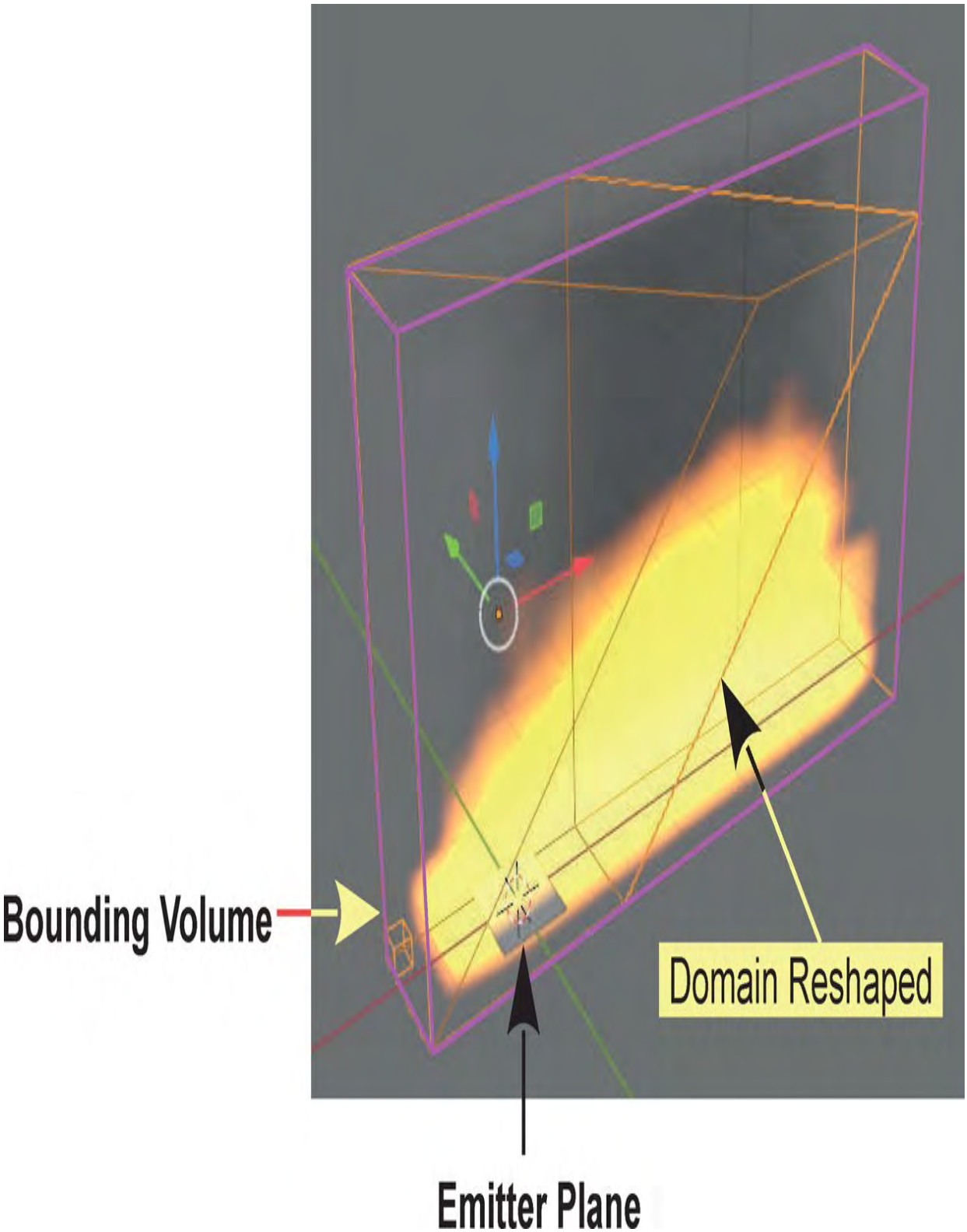


Figure 25.106

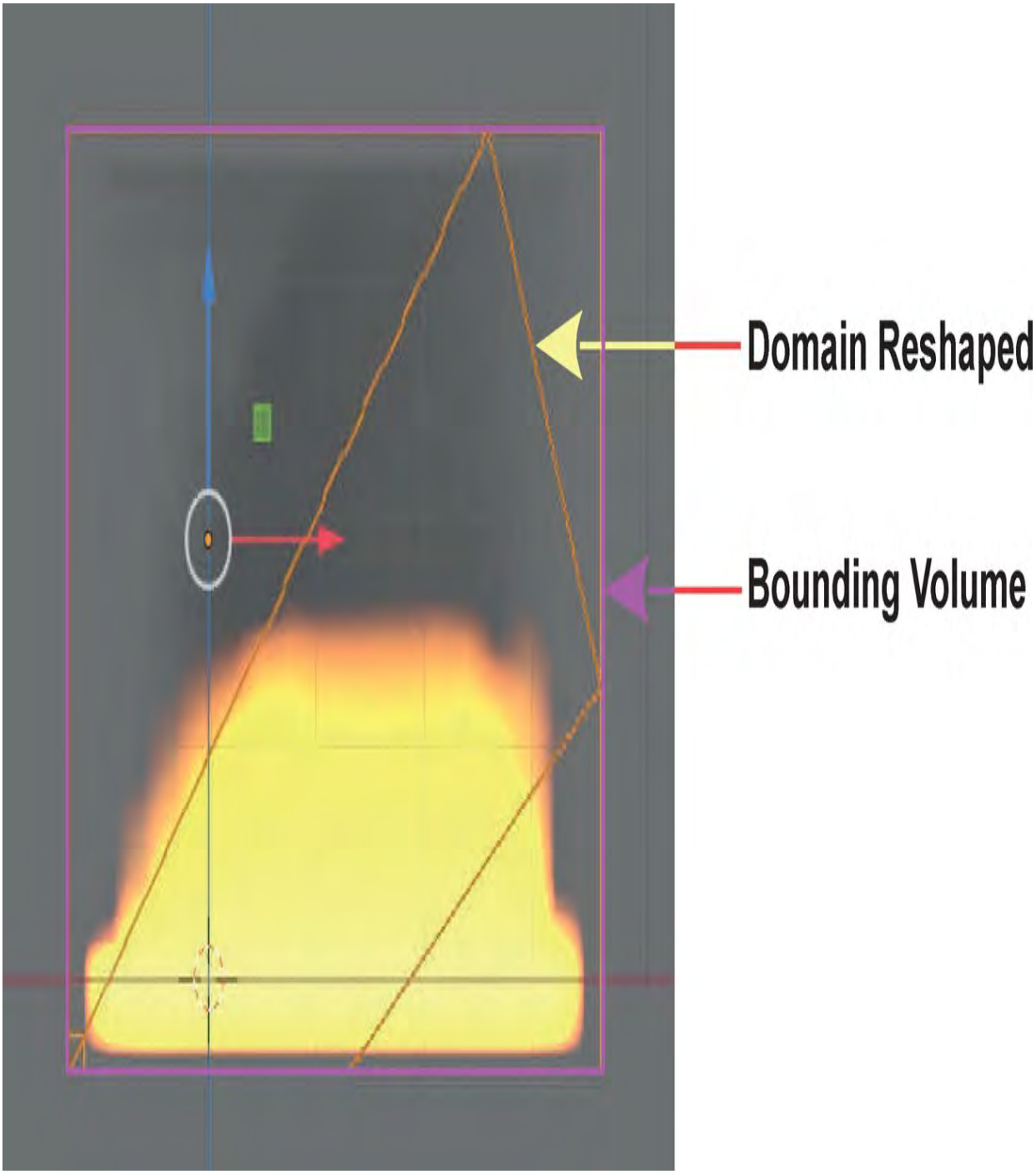


Figure 25.107

This display may or may not be what is required to suit the Scene being created.

To confine the Smoke and Fire Emission to the Emitter Object, check **Adaptive Domain** in the Physics Properties for the Domain.

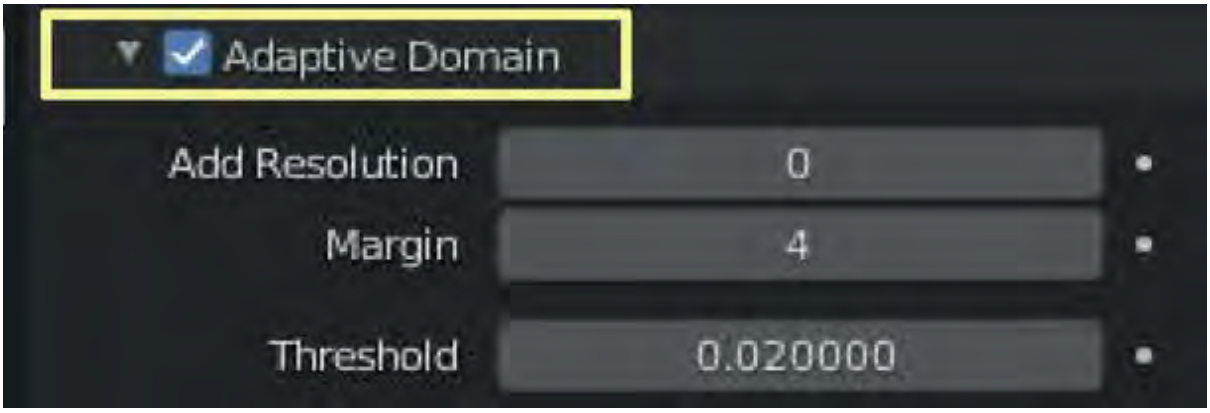


Figure 25.108

With Adaptive Domain checked the Domain, at first consolidates to the initial volume of Smoke and Fire then expands to suit the expanding volume.

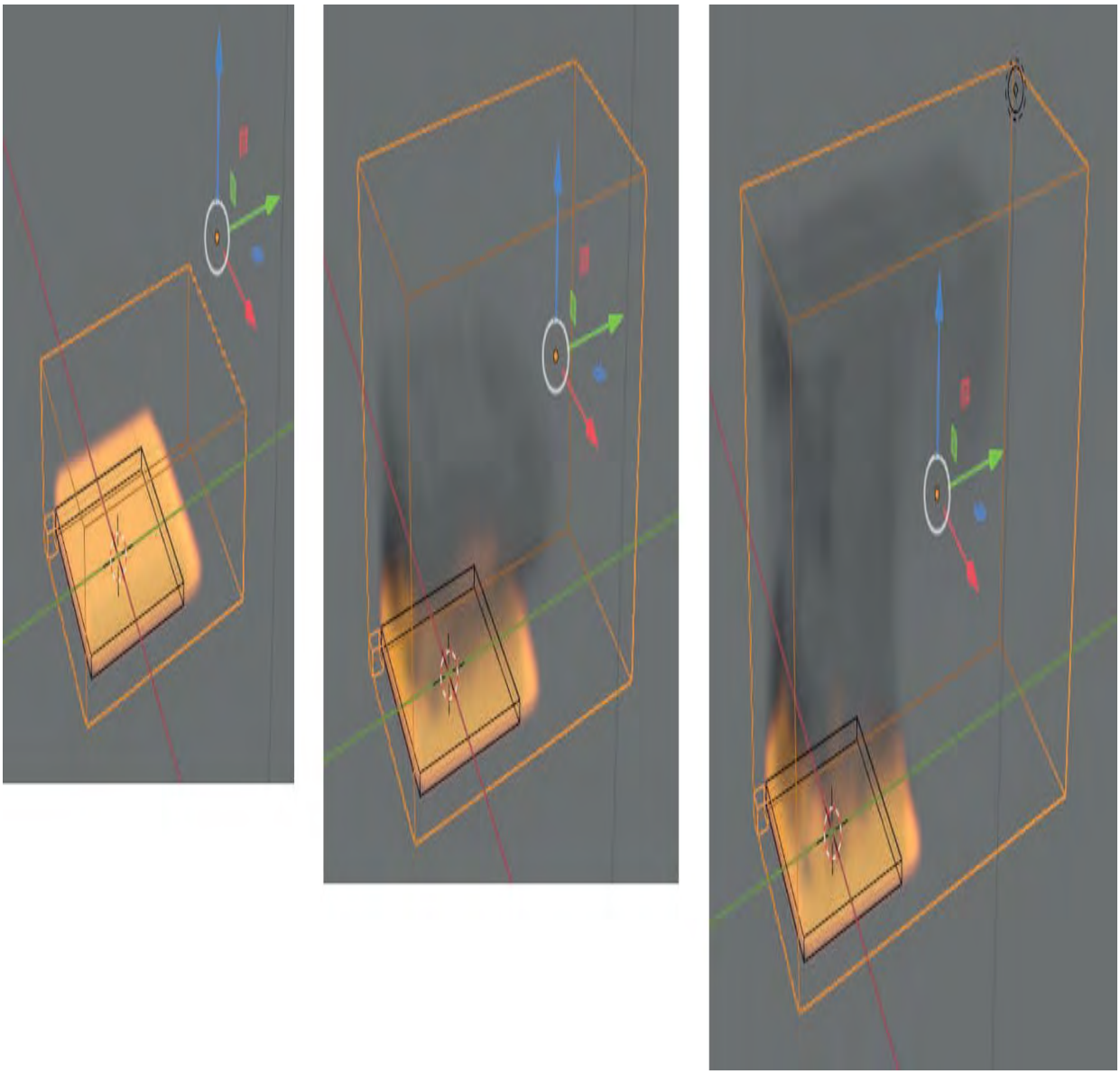


Figure 25.109

With the Domain resized and reshaped and with Adaptive Domain active a Force Field may be introduced to affect the Smoke and Fire (see Chapter 24—24.14).

3D Viewport – Right Orthographic View

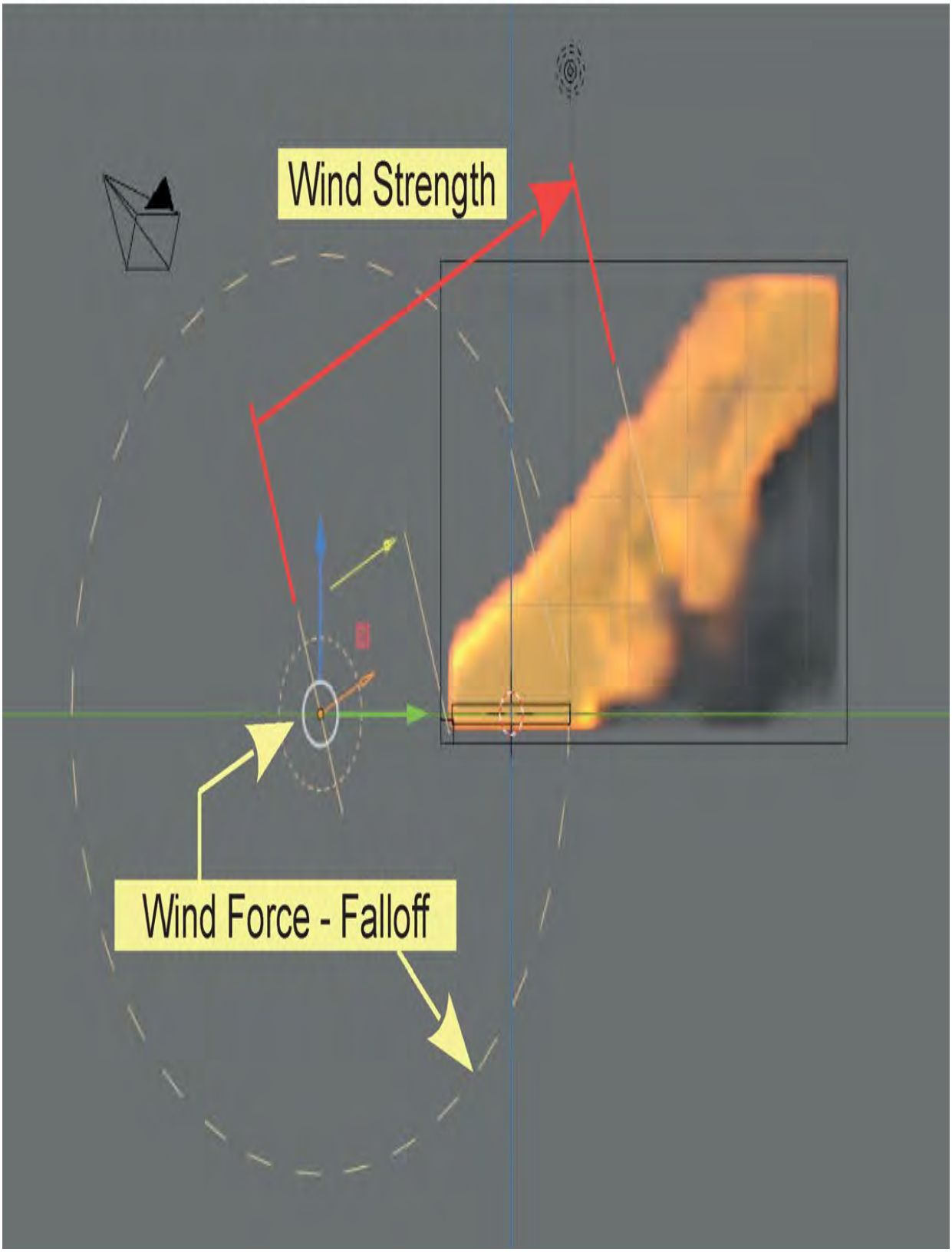


Figure 25.110 Viewport — Right Orthographic View

Note: With a Force Field active the display in the 3D Viewport Editor may hesitate when the Animation is played in the Timeline. Depending on the computer, it may take a second to perform the calculation. Figure 25.110 shows the Scene at Frame 217 when the default 250 Frames are employed.

Turn **Gravity** off to see the affect take place immediately the Animation is played.

Be prepared to experiment with settings to achieve the desired result.

26 Dynamic Paint

26.1 Dynamic Paint

Introduction

Dynamic Paint is the process of using one Object to color (paint) or deform the surface of another Object.

When coloring the process is much like painting on a canvas using one Object as a **Brush.** The Object being painted is the **Canvas.**

Although named **Dynamic Paint** the process can also deform the surface of an Object by displacing Vertices in a permanent displacement or by simulating a wave formation as one Object moves through the surface of another in a dynamic effect.

The following will demonstrate the basic setup when using Dynamic Paint and is intended to show you the method of setting the Physics for the basic operation. By adjusting settings in the Properties Editor, Physics Properties and creating Node arrangements in the Node Editor the final outputs are infinite.

Caveat: The following procedure has been developed through experimentation and describes one method only.

Understanding the procedure involved in creating a Dynamic Paint effect will allow you to research and follow detailed tutorials.

26.1 Dynamic Paint — Painting

To demonstrate **Dynamic Painting** a UV Sphere Object will be used as a **Brush** to paint a Material Color onto the surface of a Plane Object, the

Canvas. You may use any Object as Brush and paint on to the surface of any Object. A Plane gives a nice flat surface on which to work.

The word Dynamic in the title refers to the fact that, in this process, painting takes place when an animation sequence is being run in the Timeline Editor. The Brush is moved on the surface of the Canvas while the animation sequence is running. This introductory demonstration will employ **Format Type: Vertex** which means that where the Brush Object coincides with the Canvas Object, the Vertices of the Canvas at the intersection, will have color applied.

Set up a Scene as shown in Figure 26.1 with a UV Sphere Object off to the side of a Plane Object. The Plane is scaled up six times and subdivided, in Edit Mode, **eighty times** producing of Vertices. The Plane has a Material color

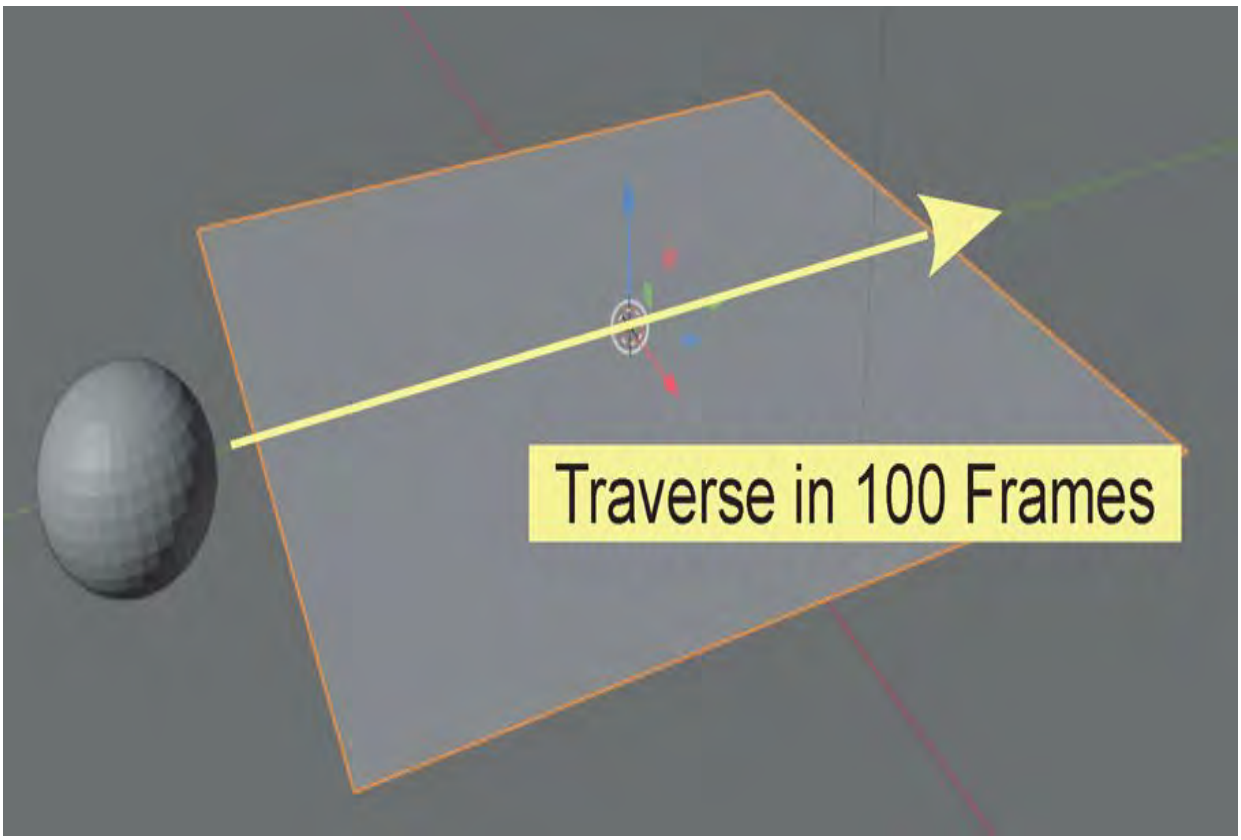
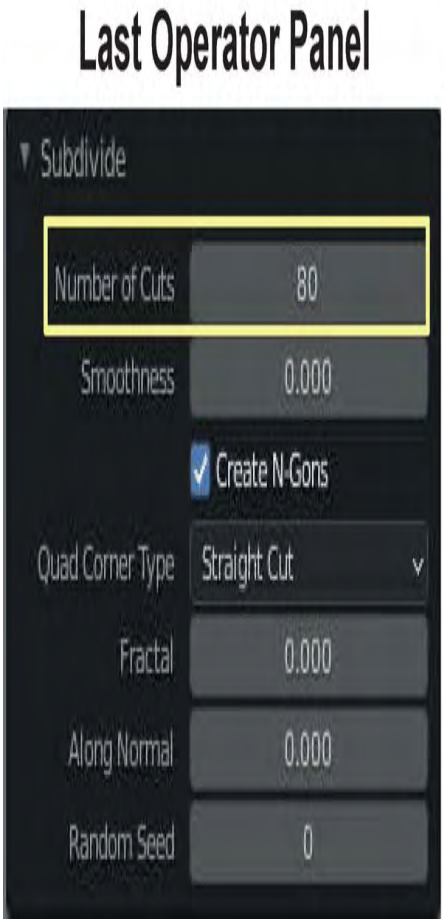
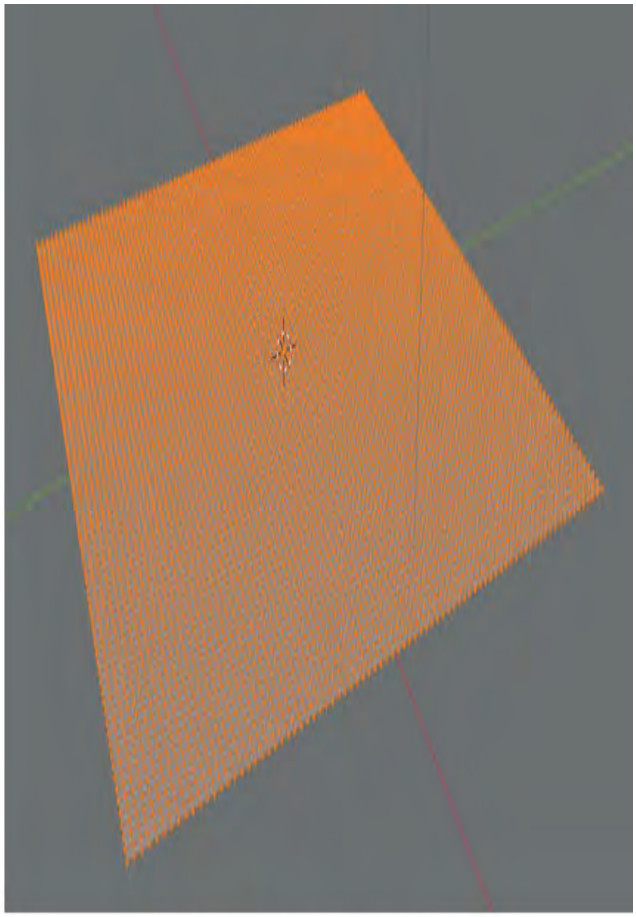


Figure 26.1

For convenience when testing the procedure, animate the UV Sphere to traverse the Plane on the Y Axis in 100 Frames.

With the Plane selected Subdivide in Edit Mode eighty times to create plenty of Vertices.





Plane – Edit Mode

Figure 26.2 Plane — Edit Mode

Plane — Edit Mode

Note: The **3D Viewport Editor** is in **Solid, Viewport Shading Mode.**

As stated in the introduction the UV Sphere Object will be used as a **Brush** to paint color on to the Plane Object which acts as a **Canvas.** The terms, **Brush** and **Canvas** are used in the **Properties Editor, Physics Properties.**

Brush Configuration

Select the **UV Sphere** (at Frame 1 in the Animation). **Note:** The Sphere **Does Not** have a Material applied but is displaying in the viewport with the default Blender gray color.

In the **Properties Editor, Physics Properties** (Figure 26.3), click **Dynamic Paint** then in the **Dynamic Paint Tab** set the **Type** to **Brush** and click **Add Brush** to display **Settings** (**Add Brush** becomes **Remove Brush**).

Figure 26.3

Make Note: Paint Color is dark blue and in the **Source Tab, Paint** is type **Mesh Volume.** These settings will be revisited later.

Viewport Shading Settings

At this stage the 3D Viewport Editor is in the default **Solid Viewport Shading Mode** with **Studio Lighting.** Since color is to be painted to the Vertices of the Plane, click in the Header to display Viewport Shading settings and change the color to **Vertex.**

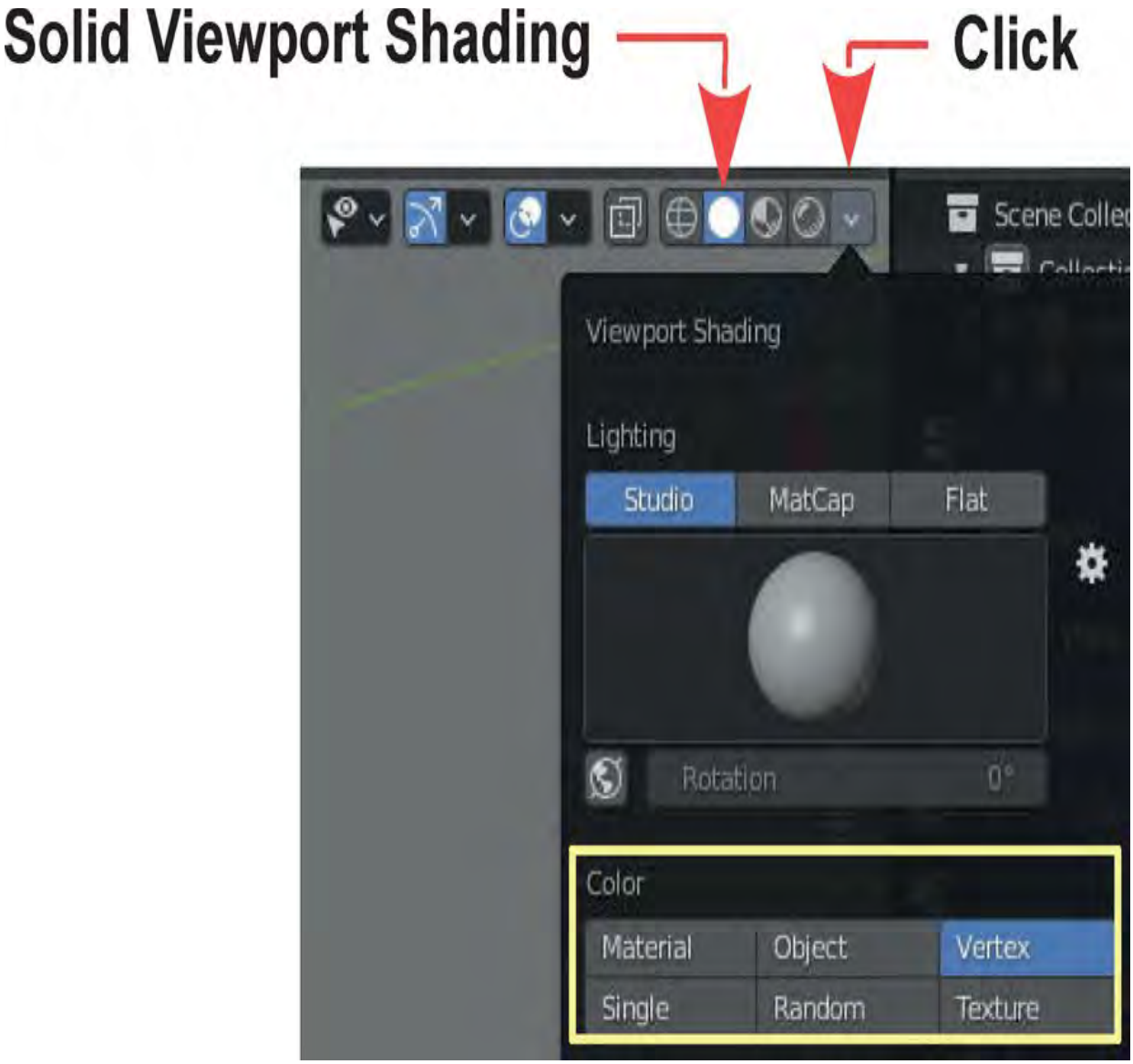


Figure 26.4

Canvas Configuration

Select the **Plane** and Tab to Edit Mode and ensure that the Mesh is subdivided with plenty of Vertices.

Tab back to Object Mode and activate **Physics, Dynamic Paint** in the **Properties Editor,** this time with **Dynamic Paint Type: Canvas** (the default). Click **Add Canvas** to display the **Dynamic Paint Properties.**

In the **Surface Tab,** make note that the **Surface Type** is **Paint.**

Expand the **Output Tab** (way down the bottom). You will see **Paintmap** and **Wetmap Layers** highlighted red (Figure 26.5).

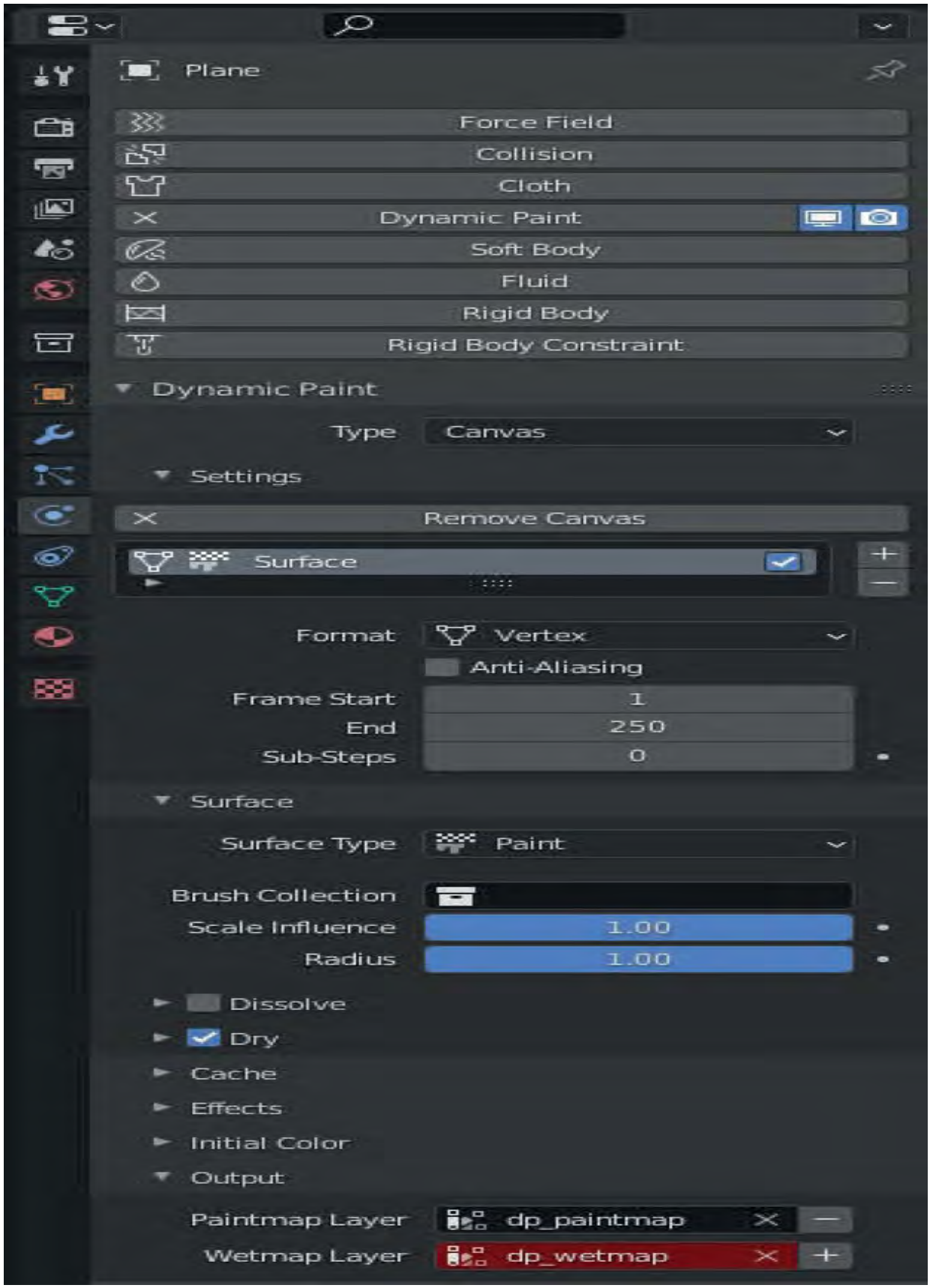

Plane
Force Field
Collision
Cloth
Dynamic Paint
Soft Body
Fluid
Rigid Body
Rigid Body Constraint
Dynamic Paint
Type
Canvas
Settings
Remove Canvas
Surface
Format
Vertex
Anti-Aliasing
Frame Start
1
End
250
Sub-Steps
0
Surface
Surface Type
Paint
Brush Collection
Scale Influence
1.00
Radius
1.00
Dissolve
Dry
Cache
Effects
Initial Color
Output
Paintmap Layer
dp_paintmap
Wetmap Layer
dp_wetmap

Figure 26.5

Click the plus sign next to **Paintmap Layer.** Clicking the plus sign changes the red highlight black

which is indicating that a paint layer has been created named **dp_paintmap.** If you click on the name you will display a panel showing **dp_paintmap** as the single entry.

The Plane in the 3D Viewport Editor displays black. **Remember** the Plane has not had a Material color applied.

Playing the Animation in the Timeline Editor shows the UV Sphere traversing the Plane creating a blue stripe. The Sphere is painting on the **dp_paintmap Layer** with the color set in the Brush settings (Figure 26.3).

Viewport and Render Displays

The display in the Viewport only shows in Solid Viewport Shading Mode with Color Vertex set in the options menu (Figure 26.4) and with Paintmap Layer, dp_paintmap active.

To have the display visible in Material Preview or Rendered Viewport Shading Modes, a Material has to be applied to the Plane using Nodes and Nodes have to be configured.

Apply Material

With the Plane selected go to the Properties Editor, Material Properties and click Add Material. By dividing the Screen and having the Shader Editor displayed you will see the Principled BSDF Node connected to the Material Output Node (the default Material).

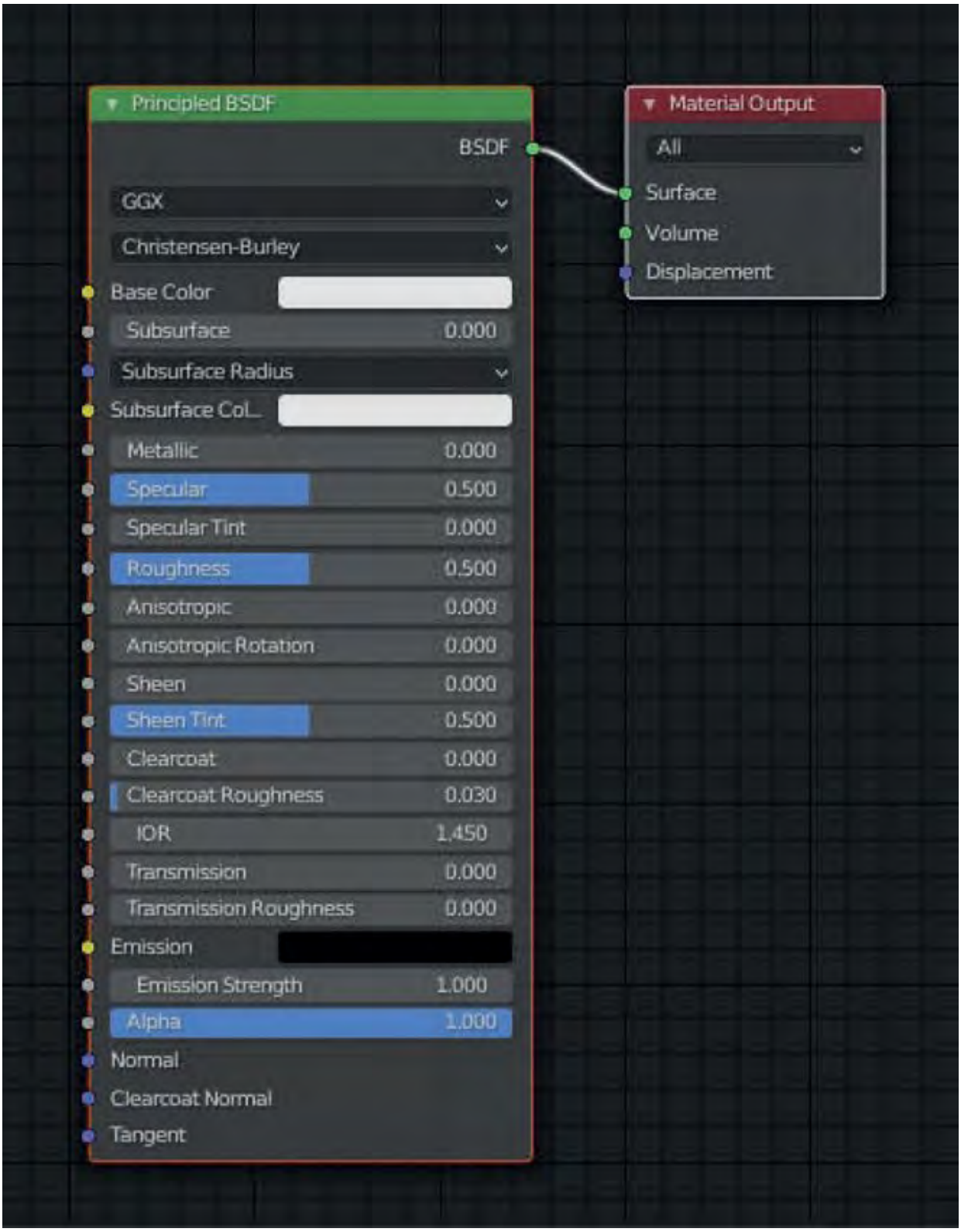

Principled BSDF
BSDF
GGX
Christensen-Burley
Base Color
Subsurface 0.000
Subsurface Radius
Subsurface Col...
Metallic 0.000
Specular 0.500
Specular Tint 0.000
Roughness 0.500
Anisotropic 0.000
Anisotropic Rotation 0.000
Sheen 0.000
Sheen Tint 0.500
Clearcoat 0.000
Clearcoat Roughness 0.030
IOR 1.450
Transmission 0.000
Transmission Roughness 0.000
Emission
Emission Strength 1.000
Alpha 1.000
Normal
Clearcoat Normal
Tangent
Material Output
All
Surface
Volume
Displacement

Figure 26.6

The Dynamic Paint Animation continues to display only in Solid Viewport Shading Mode.

Reconfigure Nodes.

Configure Nodes in the Shader Editor

Add and connect Nodes as shown in Figure 26.7 (see Chapter 16).

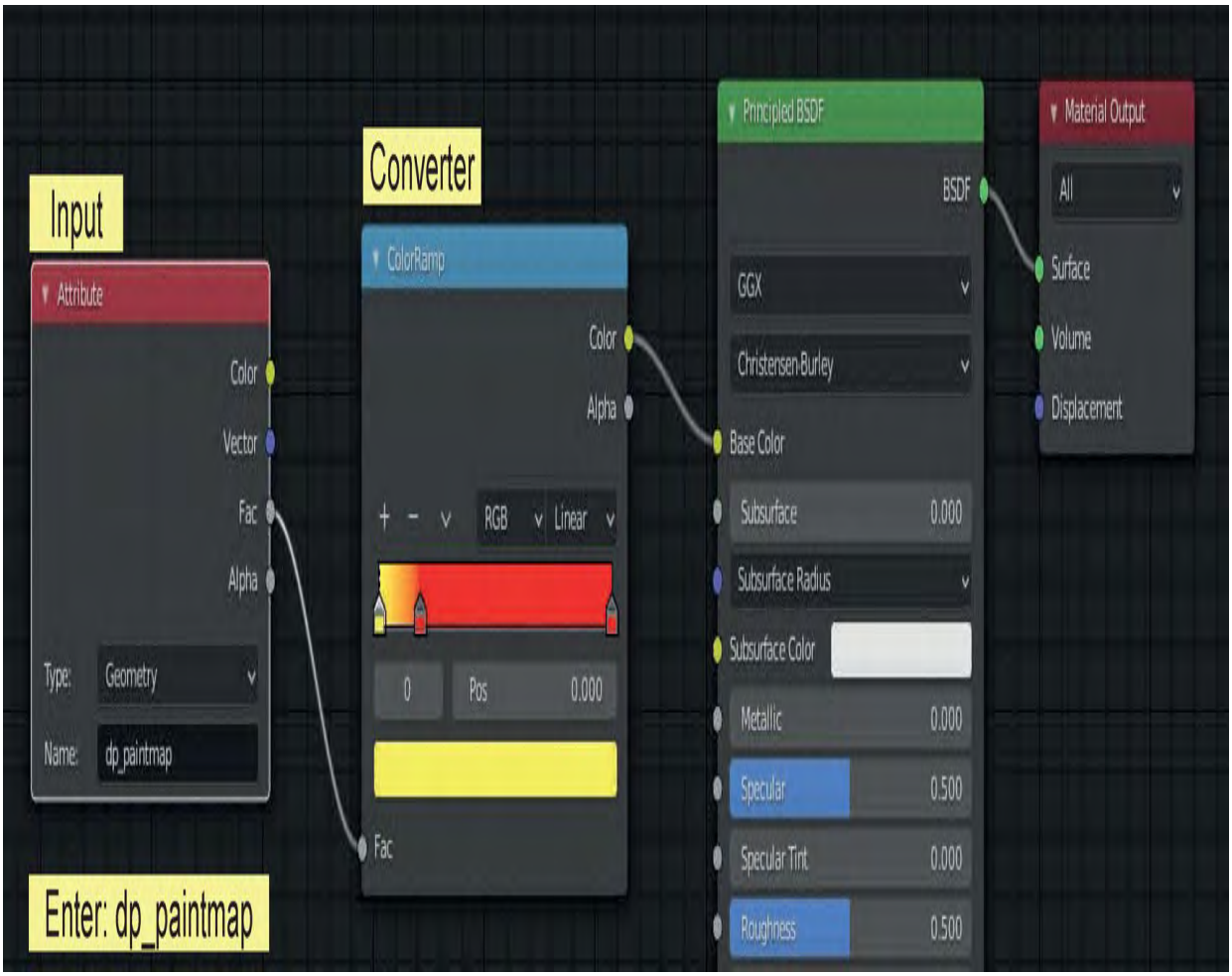


Figure 26.7

With the Nodes configured, **in this particular instance,** the Plane displays yellow and when the Animation is played in the Timeline a red stripe is painted where the UV Sphere coincides withthe surface of the Plane (Figure 26.8)

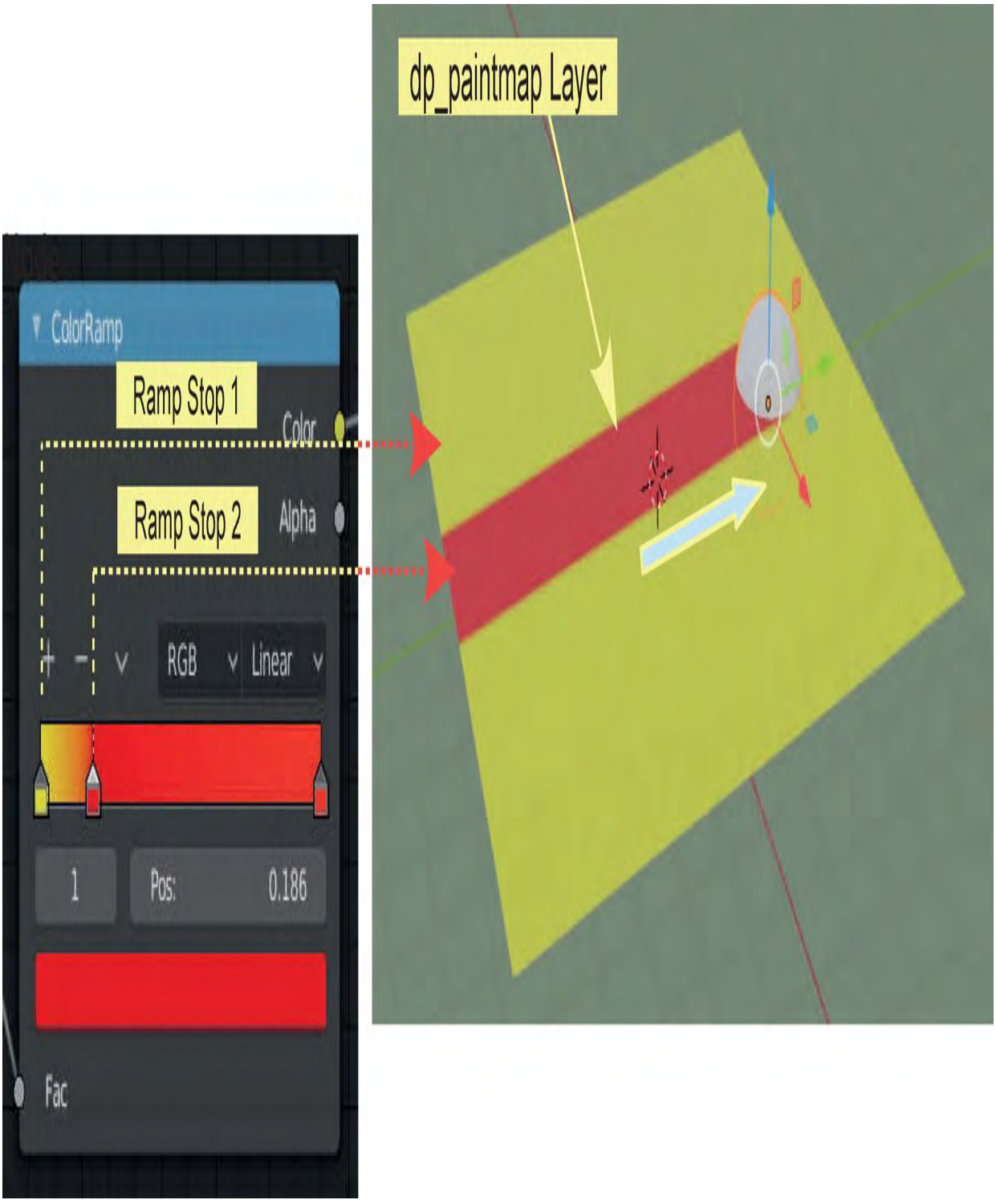


Figure 26.8

The color of the Plane and the Paint Color are determined by the settings in the **Color Ramp**

For Color Ramp settings see Chapter 16 — 16.25

With the Nodes configured the Paint Animation displays in Material Preview and rendered Viewport Shading Modes. In Solid Viewport Shading the Plane continues to display black with a blue stripe.

Brush Paint Source Options

Paint Color: In this particular arrangement the **Paint Color** for the Brush, in the Properties Editor, Physics Properties, Dynamic Paint, Settings is superseded by the Color Ramp. However, the Paint Settings in the Brush (UV Sphere) **Source Tab** have an influence.

Brush Paint Source Options

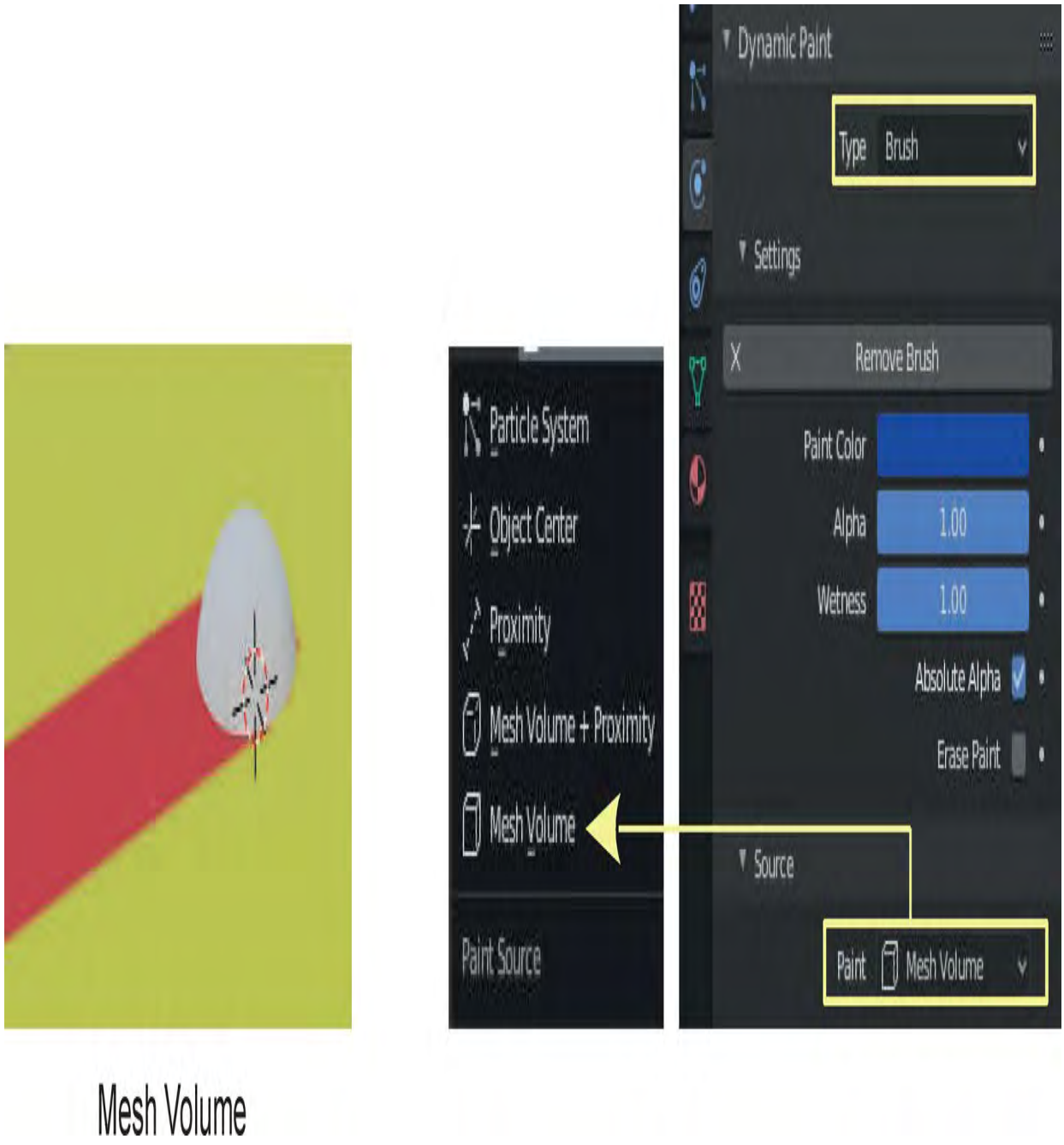
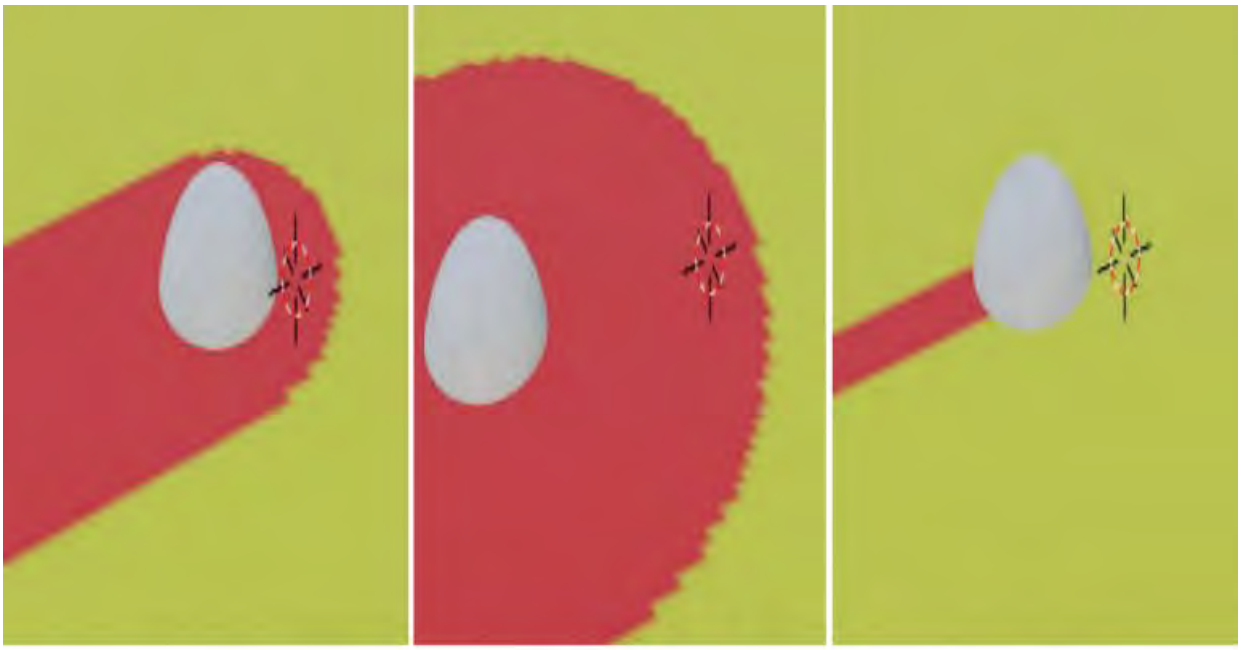


Figure 26.9

Paint Settings in the Dynamic Paint, Source Tab.



Volume + Proximity

Proximity – Distance:3.400

Object Center – Distance:0.500

Particle System

Brush Settings - Source Tab

Figure 26.10

Note: With **Particle System** a Particle System has to be created for the UV Sphere in the Properties Editor, Particle Properties and entered in the Brush Settings, Source Tab. When the Animation is played Particles Emitted from the Brush (UV Sphere) contact the Plane causing the application of color.

Tricks of the Trade

Brush Paint: ParticleSystem with the Brush elevated above the Canvas Plane (Raindrop Effects).

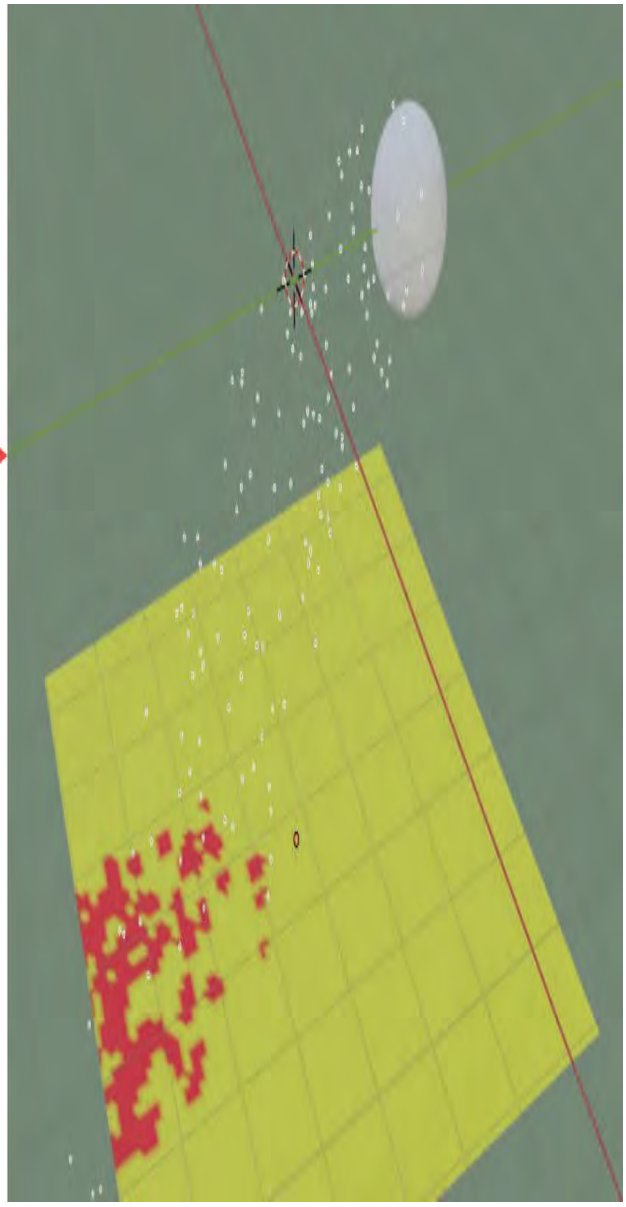


Figure 26.11

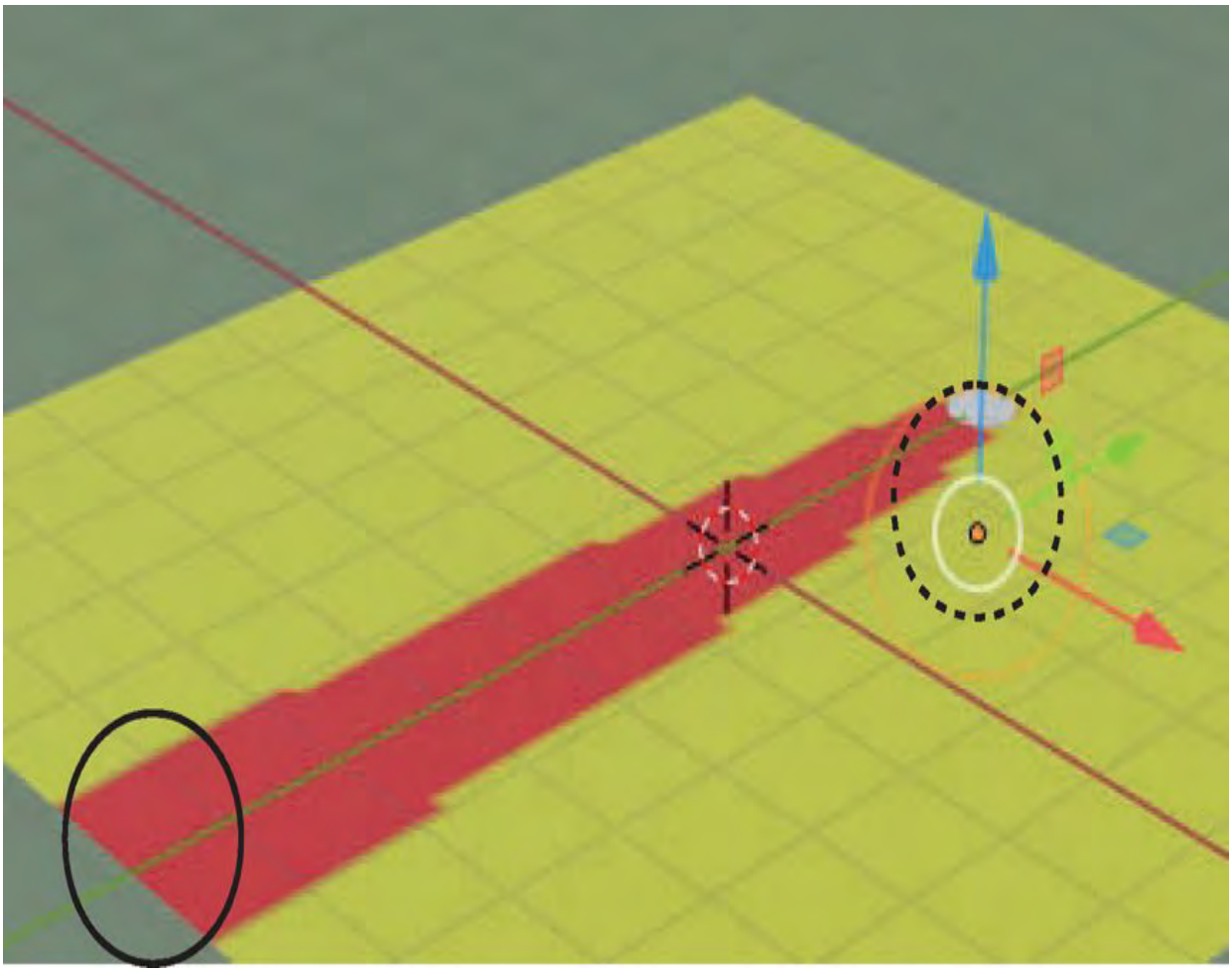


Figure 26.12

Freehand translate the Brush while playing the Animation in the Timeline or Add a Follow Path Constraint with a Curve Path.

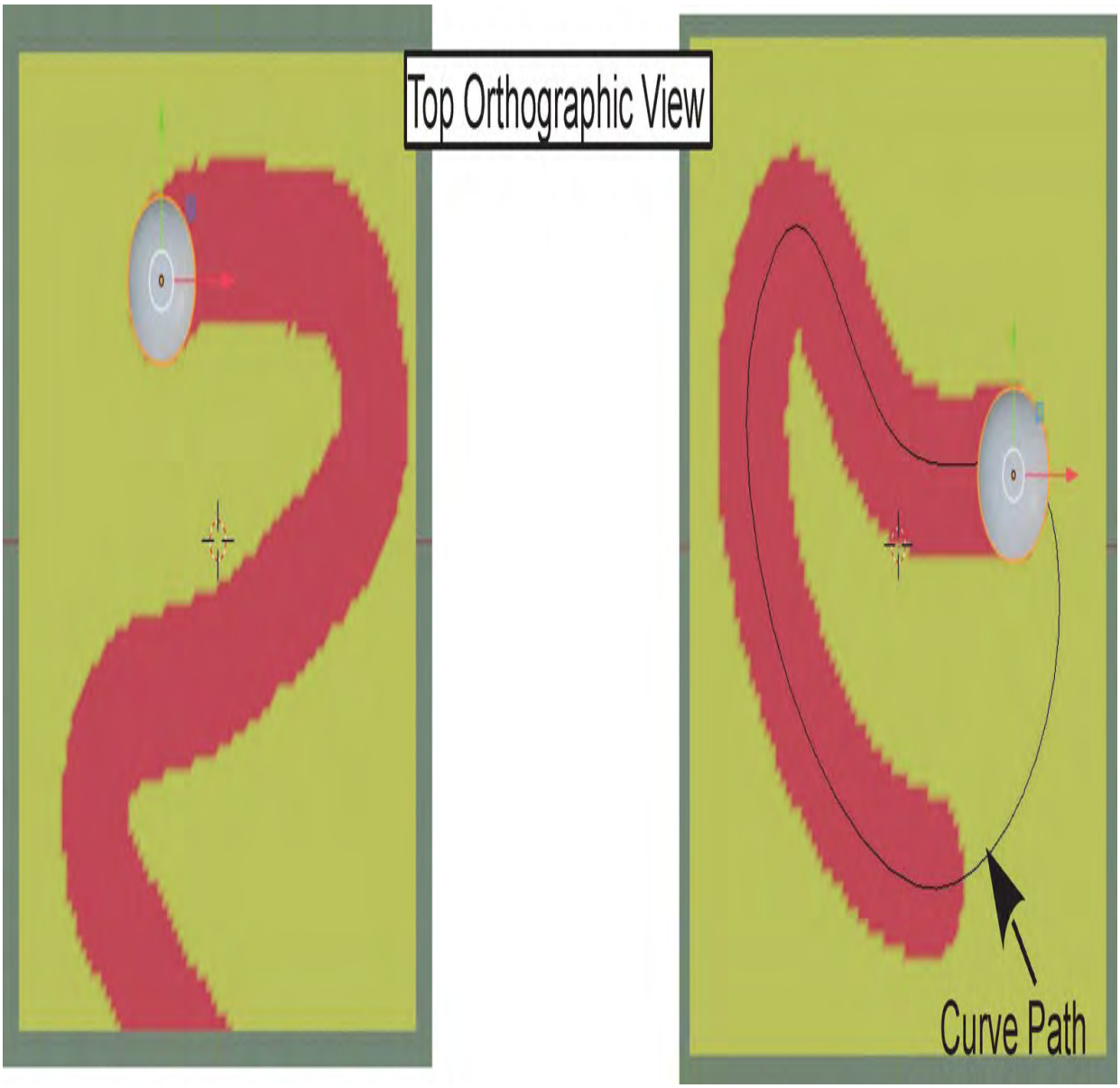


Figure 26.13

Canvas Deformatior

The Brush may also be used to deform the surface of the Canvas. Wave and Displace Options are found in the **Dynamic Paint Physics** buttons for the Canvas in the Surface Tab, **Surface Type.**

VVith **Surface Type: Waves** or **Displace** set for the Canvas there is no painting as such taking place in the simulation, therefore, the Node System used on the previous demonstration and the Paintmap Layer are obsolete. If

you have used a Particle System it is also obsolete. Changing the Surface Type for the Plane from Paint to Waves or Displace removes the Paintmap Layer from the Settings. The Node System remains active but the Attribute Node (renamed dp_paintrnap) no longer has any affect. The Color Ramp Node still controls the Material color for the Plane when viewed in Material Preview or Rendered Viewport Shading Modes but you could cancel Nodes altogether and set colors for viewing in Solid Viewport Shading Mode.

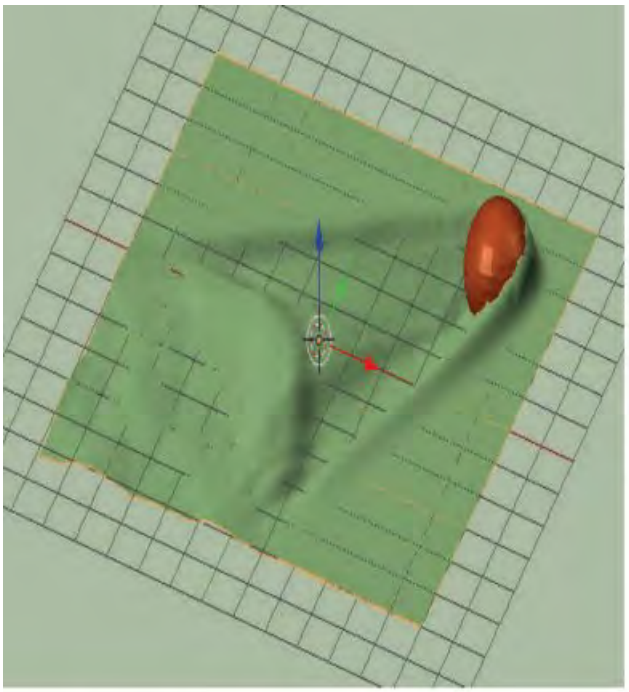
You may leave the UV Sphere Animated to traverse the Plane or you could delete the Keyframes for the Animation altogether. Be aware that an Animation is required to perform the Wave or Surface Displacement. Reset the Animation End Frame to 250 Frames in the Timeline.

O.K. With the Plane selected and Surface Type: Waves or Displace set, deselect the Plane. Press the G Key to have the Sphere in Grab Mode. With the Mouse Cursor in the Timeline press the Spacebar to start the Animation playing (yes the Animation will play without any Keyframes set for an Object).

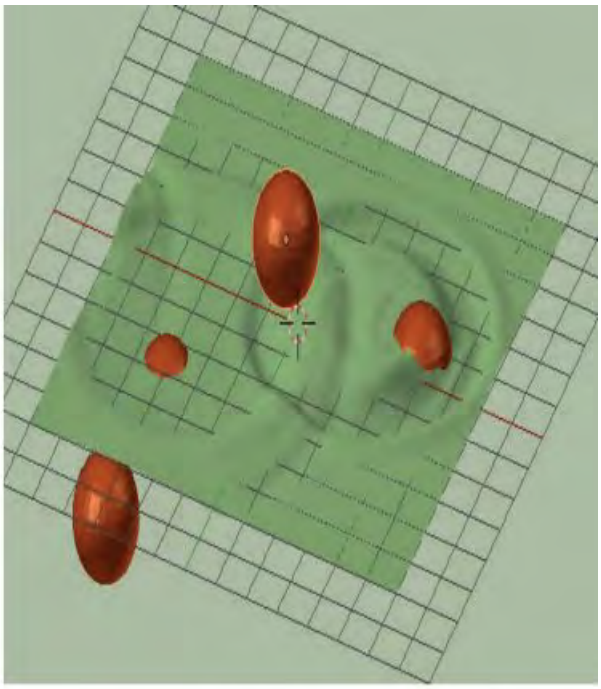
As the Animation plays, select the UV Sphere in the 3D Viewport Editor, press the G Key for Grab Mode and drag the Sphere over the surface of the Plane to see either Waves or Surface Displacement.

Wave Surface spIacernein

Surface Displacement



Sphere Dragged through the Surface



Sphere Dragged from below surface then back into the Surface

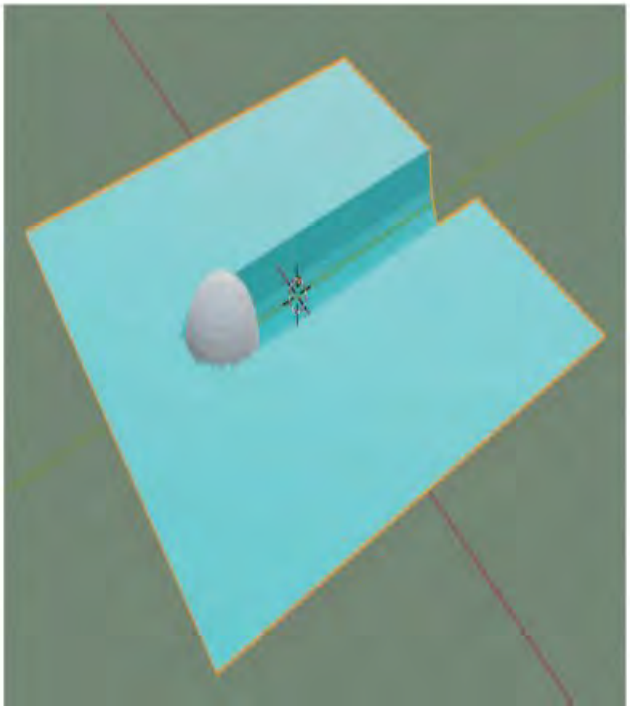


Figure 26.14

Note: The surface displacement only occurs when the surface of the Sphere coincides with the surface of the Plane.

27 Installing Add-Ons

27.1 Blender Bendy Bone Add-On
27.2 Obtaining the Add-on File
27.3 Installing the Add-On
27.4 Using the Add-on

Introduction

Add-ons are additional Blender functions which you activate when required. Blender comes with a variety of Add-ons hidden away in the **Preferences Editor** and there are literally hundreds available for download from the Internet. Add-ons are pieces of computer code written in the **Python** programming language (Python Scripts).

The Blender website contains a link to the scripts repository where a great number are found.

When you download an Add-on (Python Script), you have to **install** it into Blender, then before you can use it, you have to **activate the script.** To demonstrate this process an example named **Bendy Bone Handles** will be provided.

This example will show you how to download the compressed (.zip) file and install the script into Blender. This is an exercise in manipulating files to add functionality to Blender.

The Bendy Bone is an Armature which is parented to a Mesh Model and used to Pose the Model similar to a conventional Armature Bone.

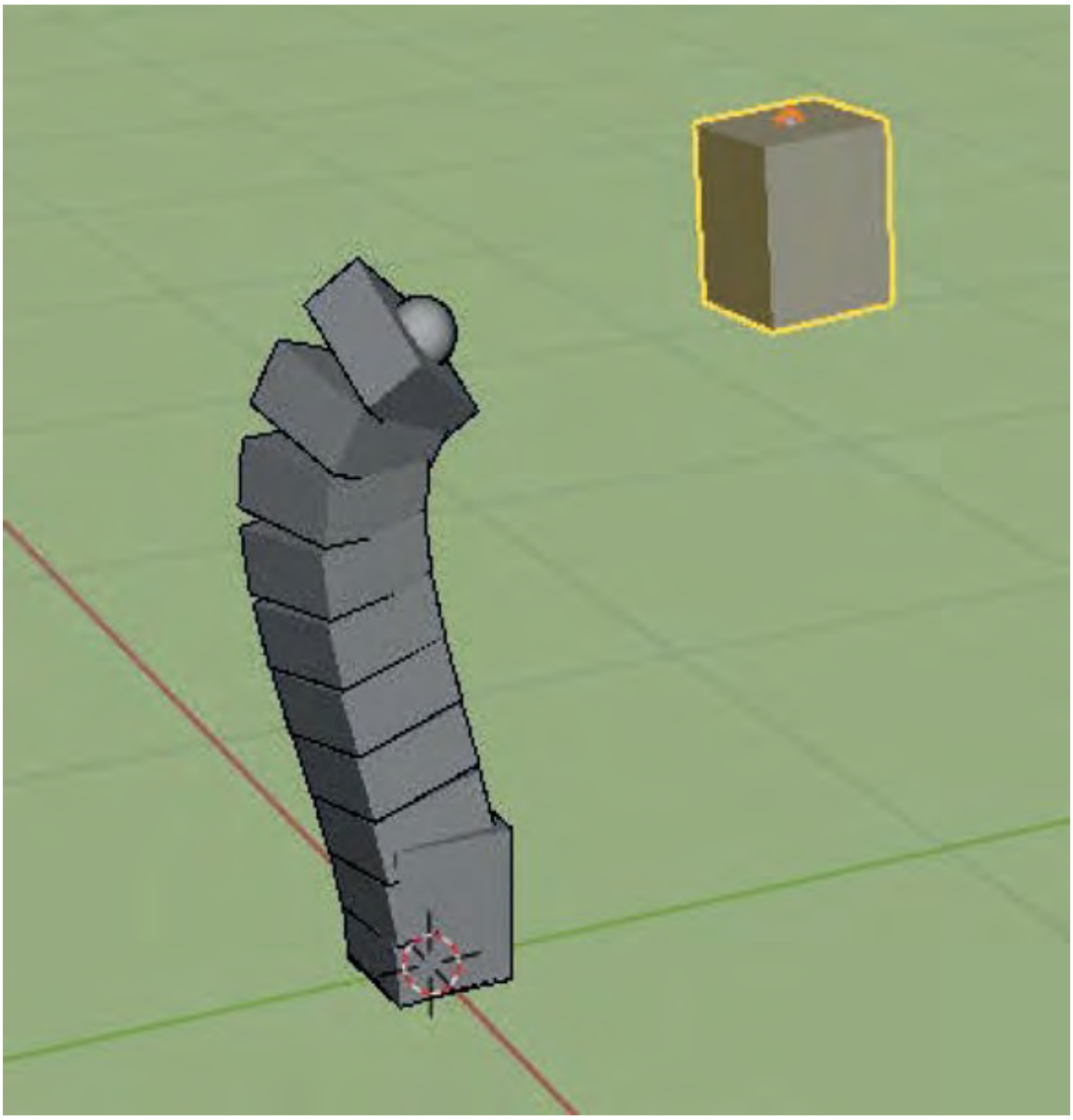


Figure 27.1

27.1 Blender Bendy Bone Add-on

The **Bendy Bone Add-on** is a special **Armature** (reference Chapter 22) which does not come preinstalled in Blender, instead, it is one of many additions for the program that are available for download. The tool is download in a compressed file (zip file) which you install into the Blender Program.

27.2 Obtaining trie Ada-on File

Go on-line, open a browser, and search for **Bendy Bone Handles Add-on - Blender**

There are several websites containing reference to the file and you will have to follow links in the websites which eventually lead you to the **GitHub** download page (Figure 27.2).

On the GitHub page for bendy_bone_handles click on **Code.** In the panel that displays click **Download ZIP** then (when on a Windows PC) click **Save File — OK** in the download panel.

gnastacast / bendy_bone_handles
Click on Code
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
master
1 branch
0 tags
Go to file
Code
Note: The GitHub page may display differently to that shown depending on the latest configuration.
gnastacast Revert "Fixed scaling issue (#4)"
d384235 on 19 May 2020
13 commits
LICENSE
Create LICENSE (#1)
README.md
Update README.md
init.py
Revert "Fixed scaling iss
README.md
bendy_bone_handles
Clone
HTTPS GitHub CLI
https://github.com/gnastacast/bendy_bon
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Download ZIP
Opening bendy_bone_handles-master.zip
You have chosen to open:
bendy_bone_handles-master.zip
which is: WinRAR ZIP archive
from: https://codeload.github.com
What should Firefox do with this file?
Open with
WinRAR archiver (default)
Save File
Do this automatically for files like this from now on.
OK
Cancel
The ZIP file will be downloaded to your PC and saved in whatever folder you have elected as you download folder. On a Windows operating system this is typically the Downloads folder.
Downloads
File
Home
Share
View
This PC
Downloads
Quick access
OneDrive
Name
Date modified
Type
Today (1)
bendy_bone_handles-master
1/08/2021 5:08 AM
WinRAR ZIP archive

Figure 27.2

Note: The GitHub page may display differently to that shown depending on the latest configuration.

27.3 Installing the Add-on

In Blender, open the **Preferences Editor** and select the **Add-ons Tab** in the left hand column. To install the new Add-on click on **Install** in the Header (Figure 27.3).

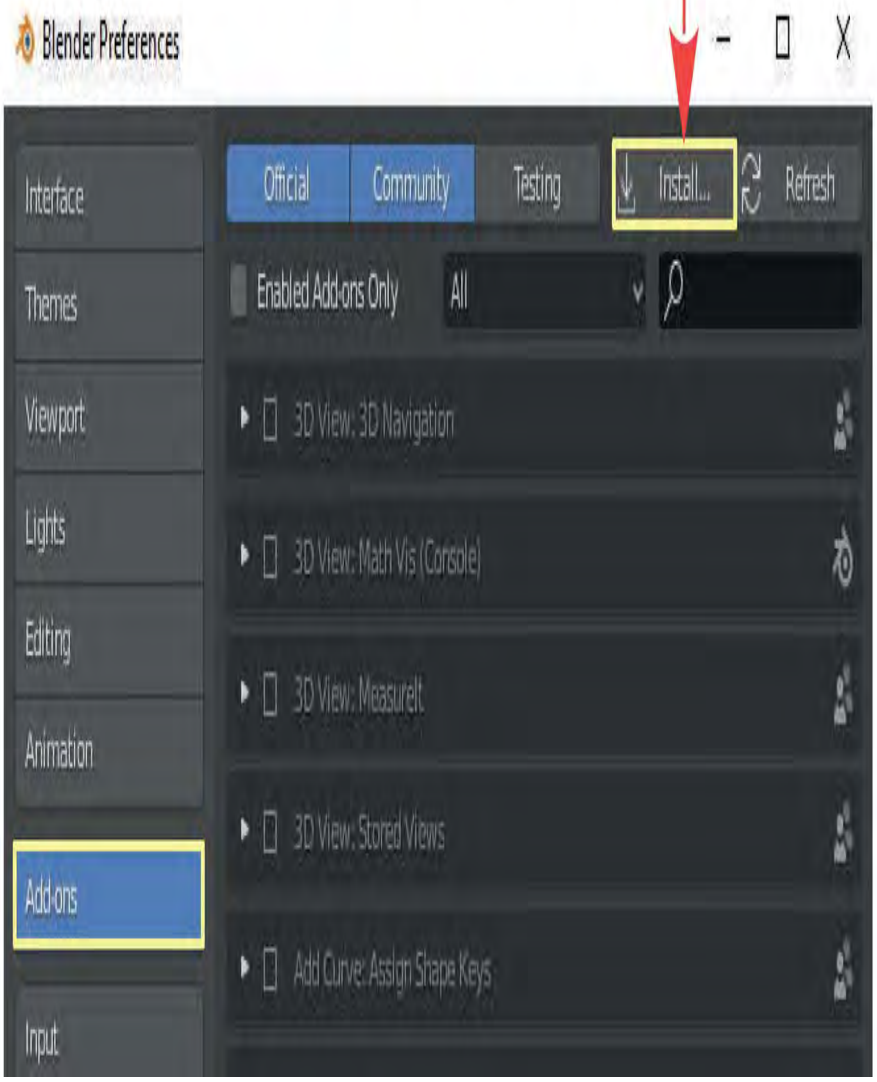


Figure 27.3

Clicking Install opens the **File Viewer.** Navigate and find the **Zip File.**

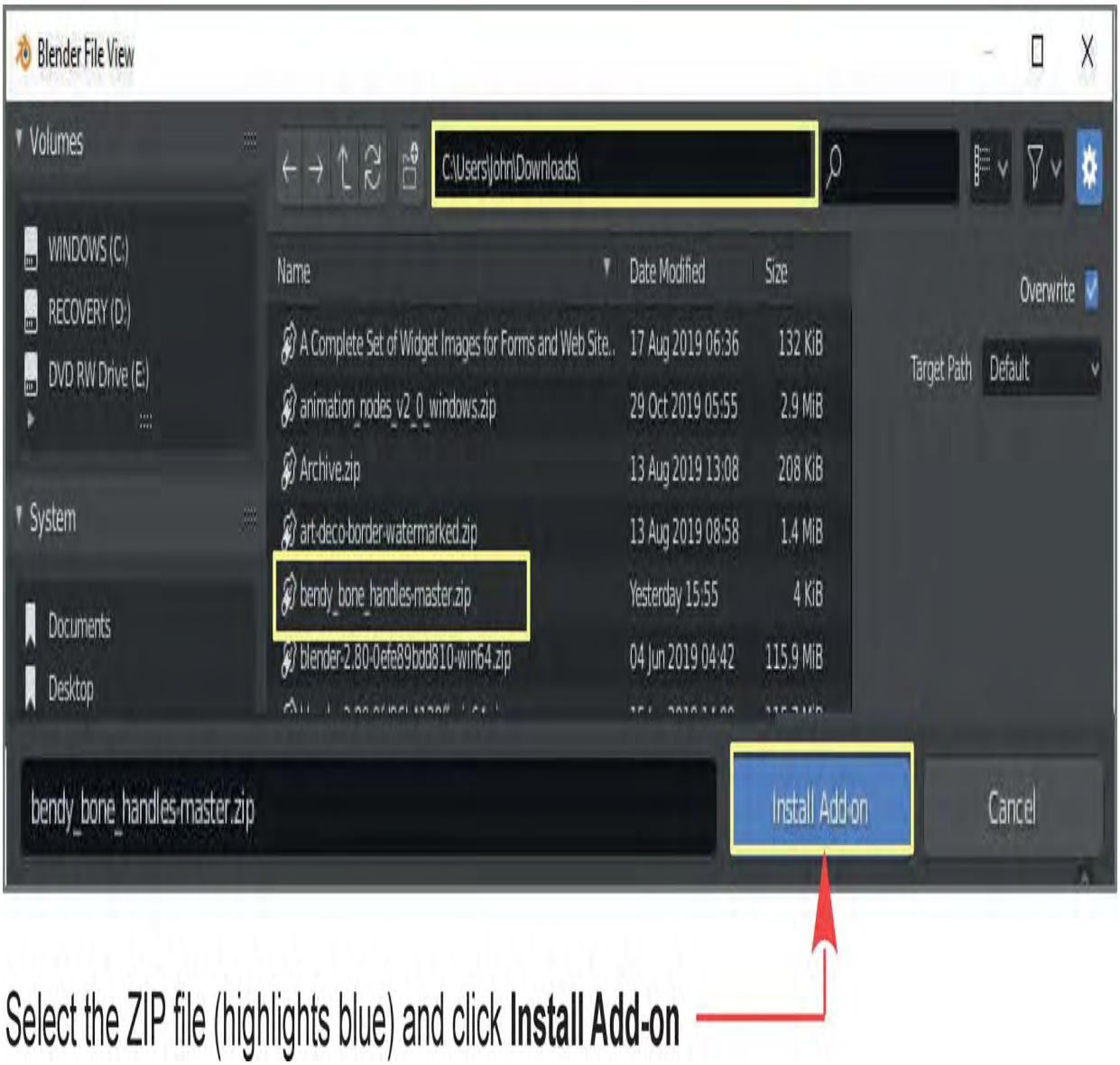


Figure 27.4

Back in the Preferences Editor you will find the new Add-on installed. Check to activate.

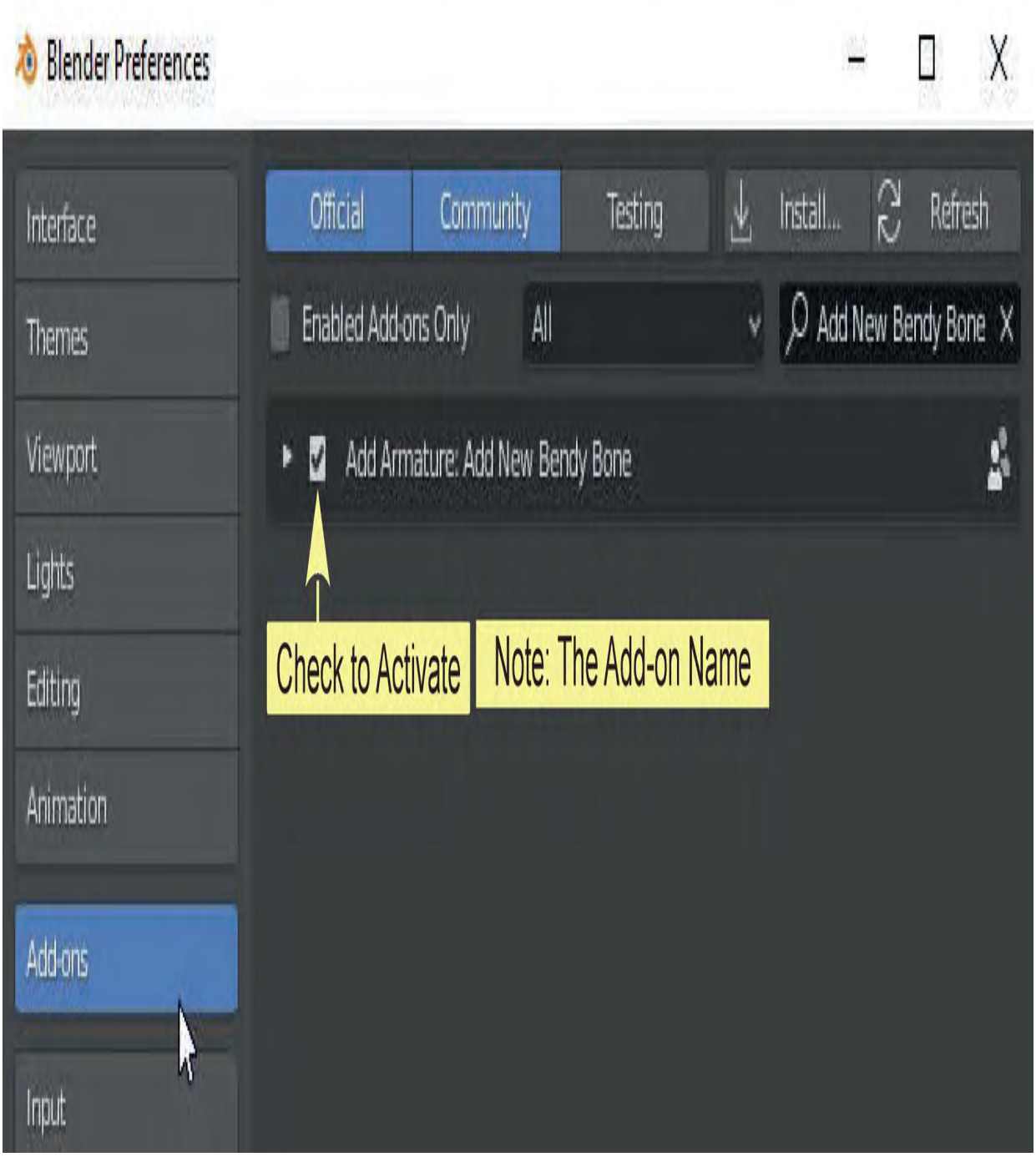


Figure 27.5

27.4 Using the Add-on

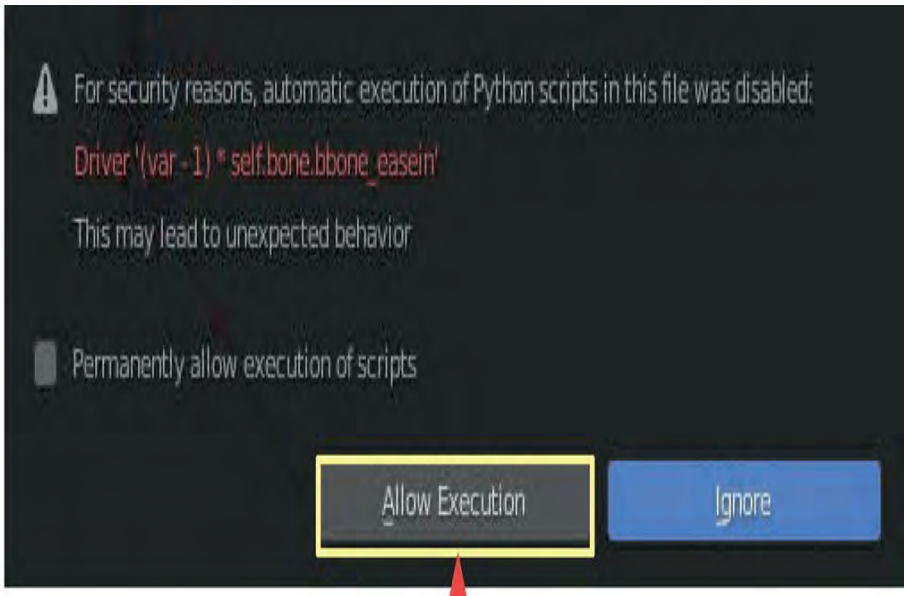
The Bendy Bone Add-on is an **Armature.** Prior to the installation of the Add-on the default Add menu contains only the Single Bone entry for introducing

an Armature to the Scene.

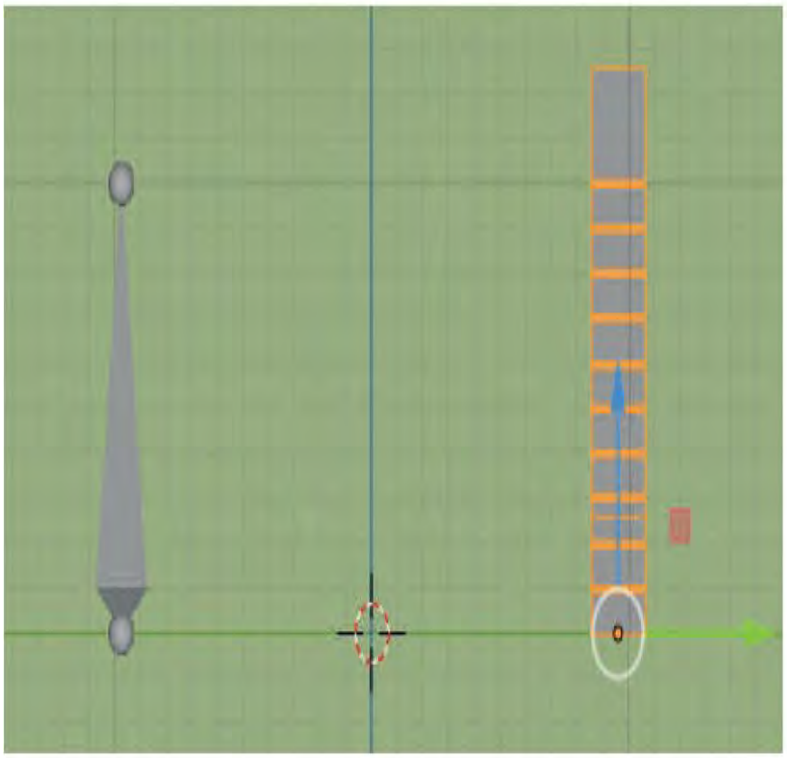
Following the installation and activation of the Bendy Bone Add-on the Add menu for Armature includes the new Bendy Bone Armature (Figure 27.6).

Figure 27.6

Electing to add a Bendy Bone armature, adds the Armature to the Scene with a Security Warning.



To use the Armature click **Allow Execution**.



Single Bone

Bendy Bone

Figure 27.7

In **Edit Mode** select the Tip or Base controls and manipulate to shape the Bone.

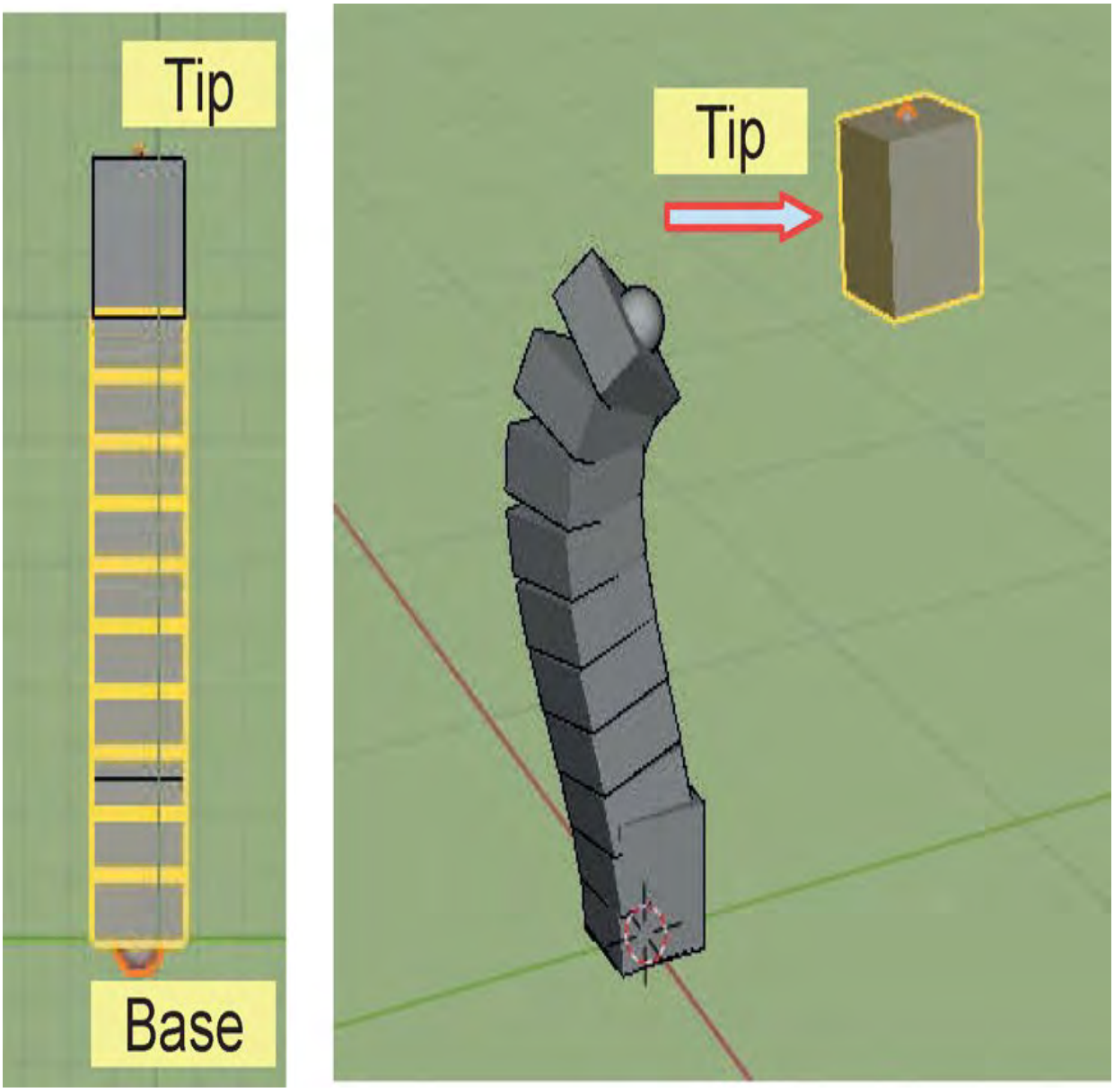


Figure 27.8

The Bendy Bone Armature is parented to a Mesh Model and used to Pose the Model as with a conventional Armature Bone.

28 Grease Pencil 20 Animatior

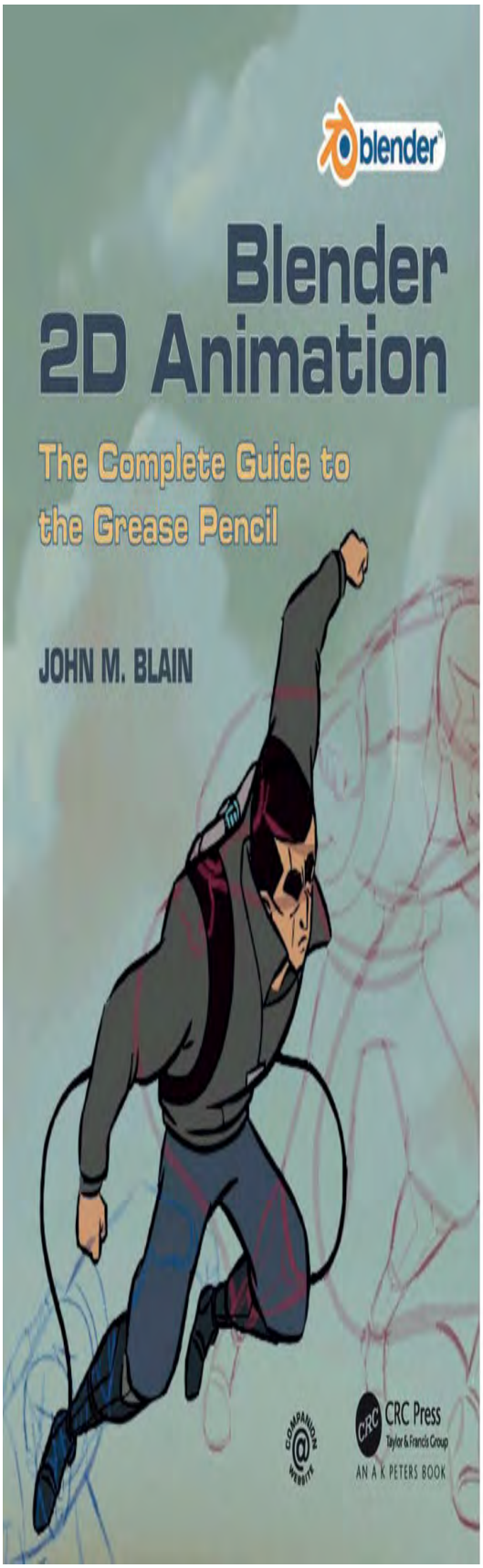
Introduction

Introduction

The Grease Pencil provides 2D Animation tools within Blender's 3D Pipeline. 2D Animation creates characters, storyboards, and backgrounds in two-dimensional environments which may be may be used in advertisements, films, television shows, computer games, or websites.

A typical Animation is a series of drawings creating a motion Cycle. Each individual drawing is captured as an Animation Frame which when played in succession creates the illusion of motion.

This subject is extensive and has, therefore, been released as a separate publication:



The Grease Pencil within Blender has its own dedicated Workspace accessed in the default Blender Screen Header by selecting **2D Animation**

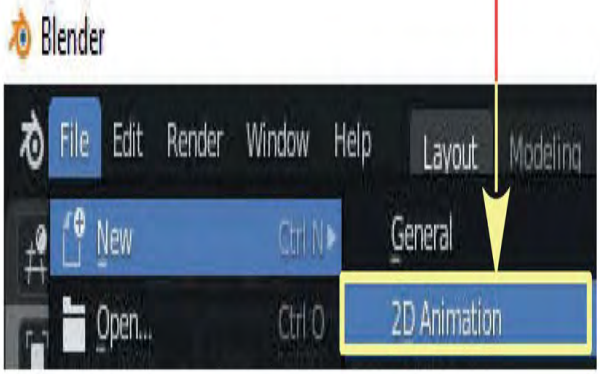


Figure 28.1

The 2D Animation Workspace is shown in Figure 27.2 as an introduction.

The Grease Pencil 2D Animation Workspace

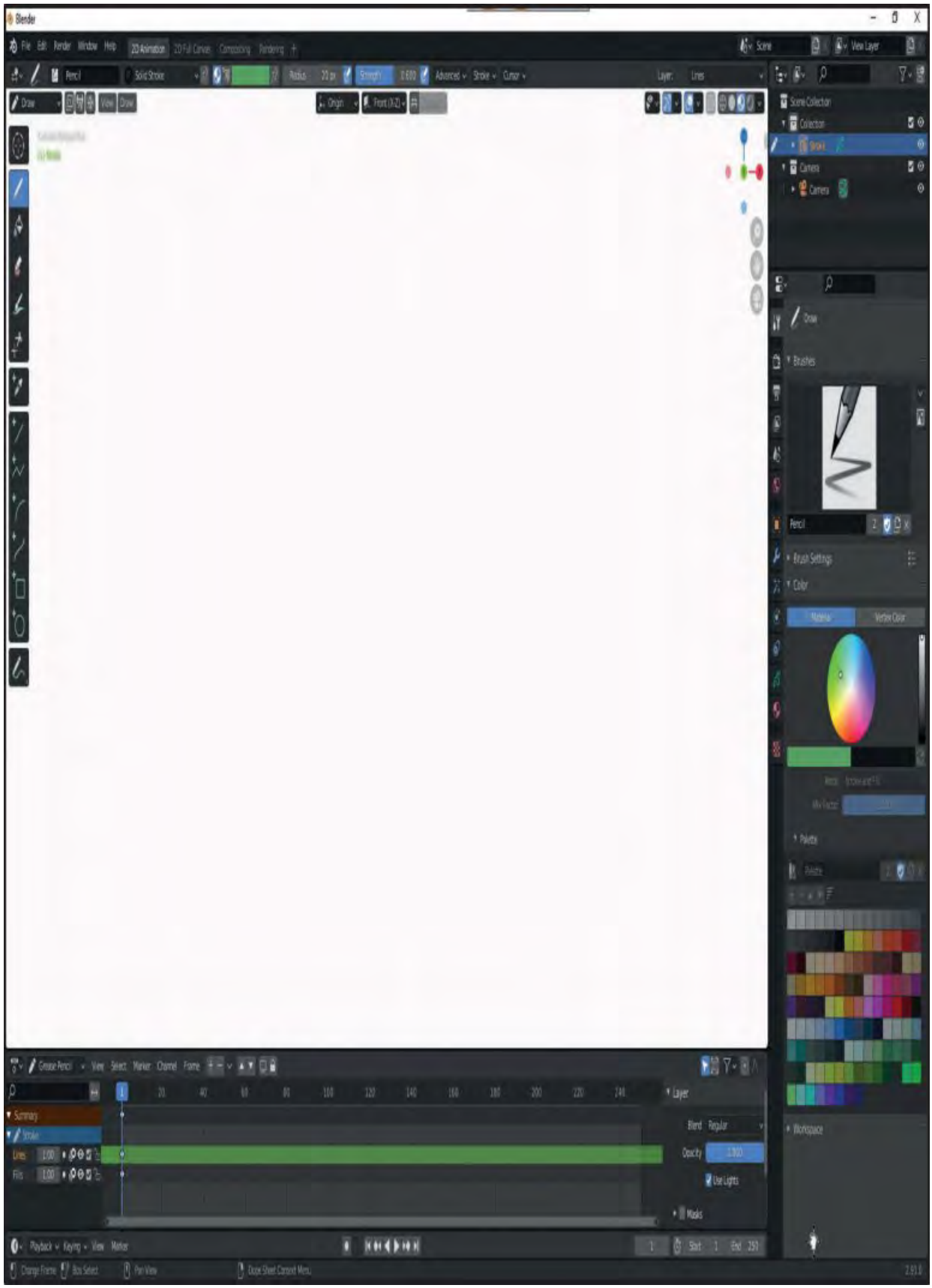


Figure 28.2

To view details included in **The Complete Guide to the Grease Pencil** with sample pages see the Author's website at:

Tamarind Creative Graphics

BLENDER
COMPUTER MODELING & ANIMATION

Books and Tutorials

Read the Author's Learning Note

Figure 28.3 www.tamarindcreativegraphics.com

29 Making a Movie

29.1 Making a Movie
29.2 Storyboard
29.3 The Video File
29.4 The Sound File
29.5 Video Editing Workspace
29.6 The File Browser Editor
29.7 Preparation
29.8 Video Sequence Editor
29.9 Rendering the Movie
29.10 Additional Features

Blender incorporates a complete Movie Making — Video Editing Workspace.

Making a Movie is performed in the **Video Sequence Editor** where you compile Video Sequences. The term **Movie** originated from Moving Pictures. Moving Pictures were originally made to entertain and tell stories and this has developed into modern Communication Systems. The basic concept however, which is, to communicate a story, remains .

Blender provides the tools which allow you to tell your story by using animated pictures (animations). You create Scenes in which actors move depicting events that you wish to communicate to an audience. The animated Scenes are recorded and **Rendered to Movie files.** The individual files are not necessarily produced in a sequence that tells the story, therefore, they need to be arranged in the correct sequence, hence the **Video Sequence Editor.**

Movies are made by piecing together short segments of video produced when you render animation sequences. Sound files and special effects are added to enhance the visual and audio presentation.

29.1 Making a Movie

Making a Movie in Blender (Chapter 19 — 19.10) will be demonstrated by producing a **Video Sequence** from a series of short animations which have been rendered to **Video Files** (Video Clips). The animations may have been created in separate Scenes in a single Blender file or in different Blender files. In either case the animations must be pre-rendered into **Video File Format** and be saved to a folder on your hard drive. The files should preferably be named or numbered in relation to a sequence of events which will tell your story.

29.2 Storyboare

A movie is a visual way of telling a story or communicating a message. To effectively piece together a movie you must have at least an idea of how you want to tell your story. In other words you should have a plan or sketch to use as a reference. The plan is called a **Storyboard.** lt is easy to become immersed in the technical detail of the process and lose the plot.

In the this demonstration, a submarine on the surface of the ocean, dives underwater and conducts a torpedo attack. The story has been broken down into five parts. Submarine on surface, submarine dives, two underwater views and firing torpedoes. Each part has been animated in a separate Scene in the same Blender file then rendered to an **.AVI video file.**

The video files are all rendered from 250 Frame animations which when combined, equals a movie of 1250 Frames. The movie will be rendered for **PAL TV** which plays at 24 frames per second, therefore, the movie will play

for approximately 52 seconds. It is a long way from being a feature film but will give you a basic idea of how a Movie is made.

To demonstrate the process of compiling a Movie work through the procedure as follows. The demonstration will combine the five video files and a sound file.

Sound file? Sound files can be background music, recorded voice, sound effects, in fact anything to enhance the video. For the purpose of the demonstration a sound file has been compiled in **.wav** format. As with video files there are many types of sound files. You are probably familiar with **MP3 and WAV** etc.

29.3 The Video Files

Figure 29.1

29.4 The Sound File

For this demonstration a series of sounds, downloaded from **Free Sounds** at **www.freesound.orq,** have been combined (Figure 29.2) using the free program **Audacity.**

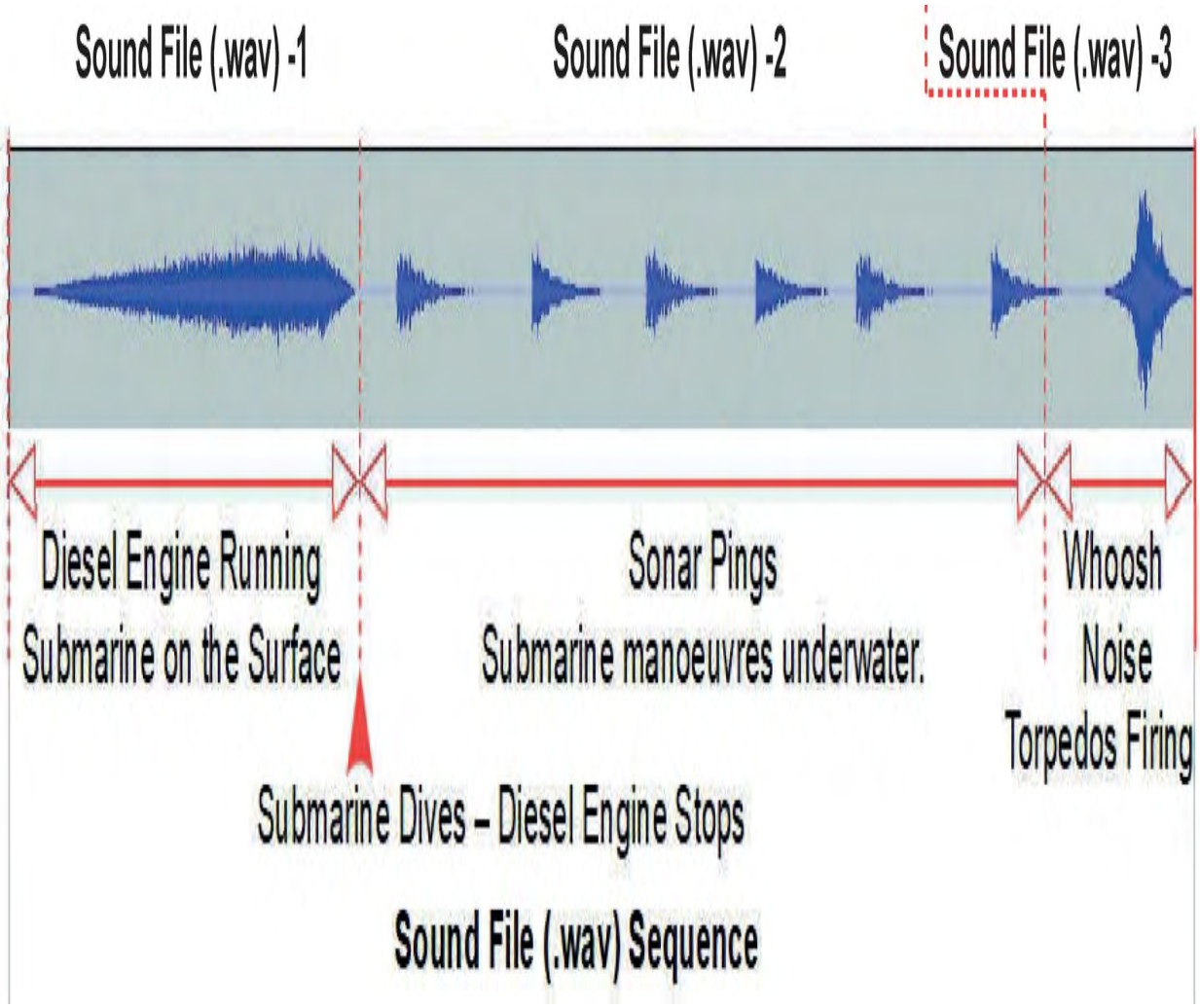


Figure 29.2

29.5 Video Editing Workspace

Video assembly is performed in the **Video Editing Workspace** hidden away. In the Screen Header. Click on **File,** then **New** and select **Video Editing** (Figure 29.3).

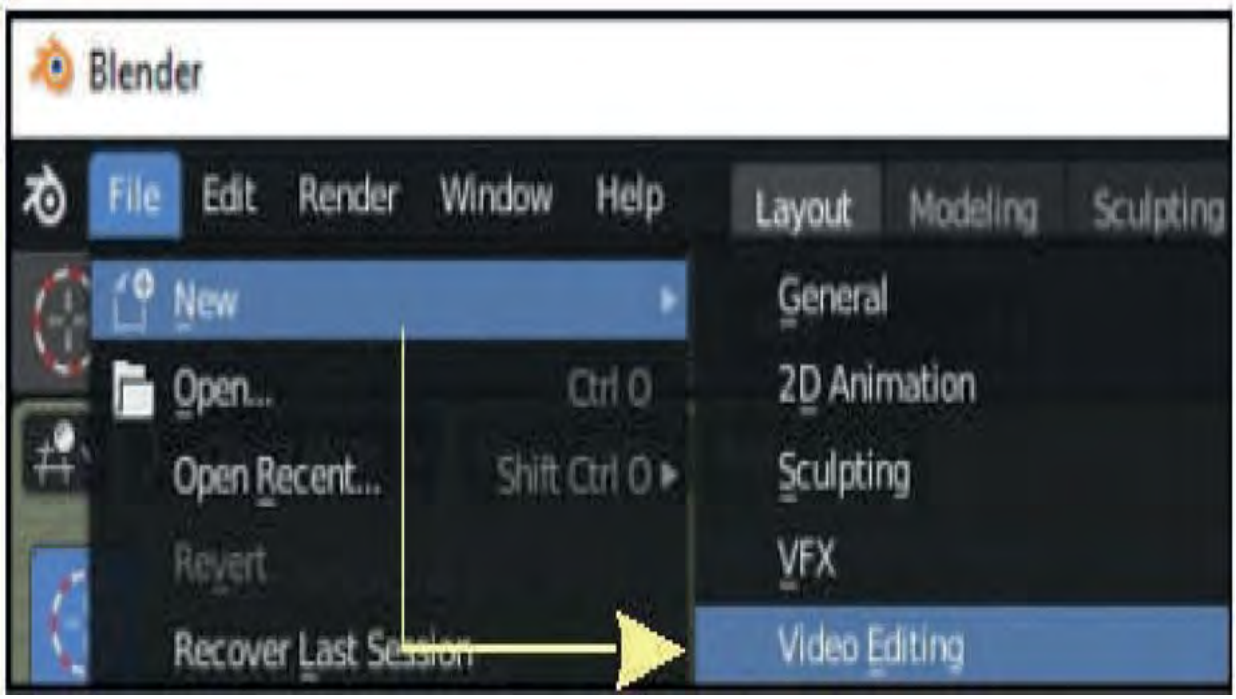


Figure 29.3

The **Video Editing Workspace** (Figure 29.4).

The **Video Editing Workspace** (Figure 29.4).

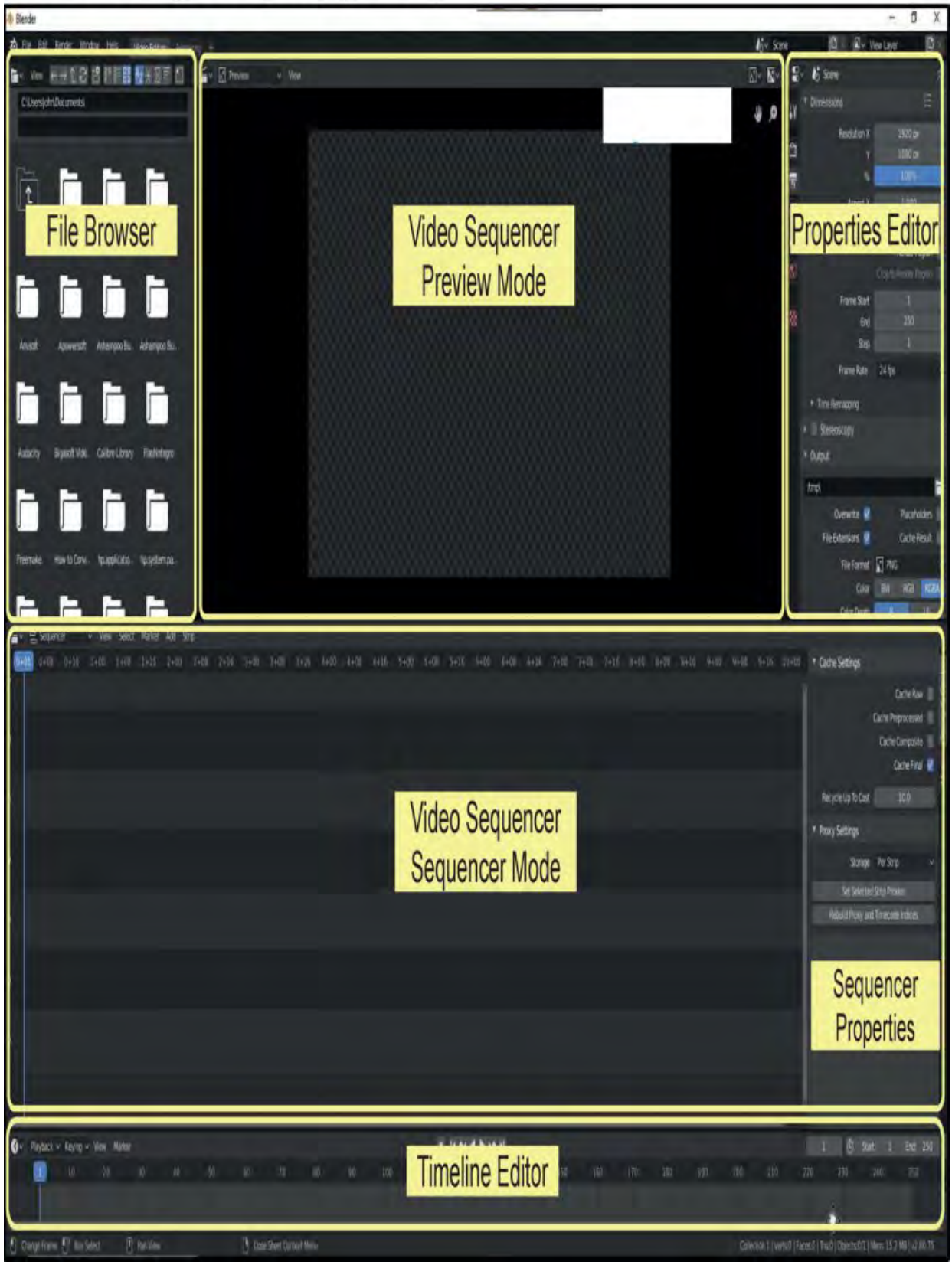


Figure 29.4

VSE Editors and Panels

File Browser Editor: Where you navigate and select files.

VSE Preview Mode: Where you see the video playback.

Properties Editor: Controls relevant to the VSE.

VSE Sequencer Mode: Where you combine Video Files (clips).

Timeline Editor: Provides control of how the video

Sequencer Properties: Video Channel Properties (Withe the Mouse Cursor in the the VSE Sequencer Mode. press the **N Key** to toggle Hide and Display)

•

In this basic instruction you will be concerned with the **File Browser Editor** and the **two versions of the Video Sequencer.**

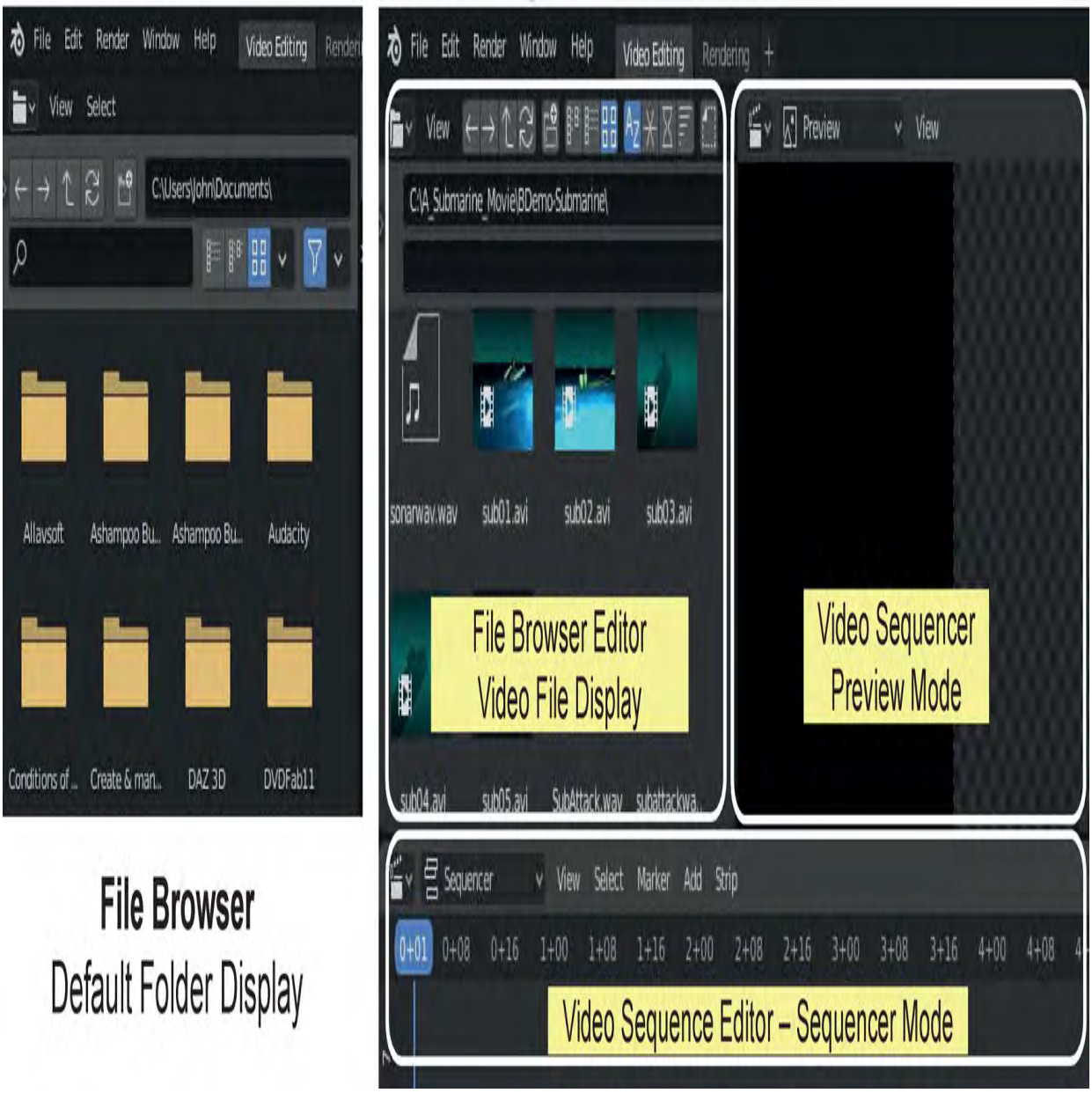


Figure 29.5

29.6 File Browser Editor

The **File Browser Editor** was discussed in Chapter 3 — 3.5 but a point to remember is, the Editor will show files in a variety of ways.

The default display in the Video Editing Workspace shows thumbnail images of a series of Folders. When you navigate to the B_Demo_Submarine Folder the first Frame in each video file is displayed. You can change this to

display the file names if you wish. Click on the button shown in the diagram (Figure 29.6).

File Browser Header
View
C:\Blender_5th_Edition_Checked\B5_027_Video_Sequence_Editor\BDem_Submarine\
sub01.avi
sub02.avi
sub03.avi
sub04.avi
sub05.avi
12-Jul-09 16:05 297 MiB
12-Jul-09 16:17 297 MiB
12-Jul-09 19:36 297 MiB
12-Jul-09 19:41 297 MiB
12-Jul-09 16:38 297 MiB
File Names
Thumbnail Images – First Frame
sub01.avi
sub02.avi
sub03.avi
sub04.avi
sub05.avi

Figure 29.6

29.7 Preparation

File Definition: In this demonstration the five **.avi files** saved to the hard drive will be referred to as **Video Files.** When combined, the final output will be called, the **Movie File.**

Before attacking the **Video Sequence Editor** some preparation, which is required, must be performed. The first step in the movie making process is to set the file path to the location where you want your **Movie File** saved and to define the **Video Output Format.**

By default, Blender sets the file path for saving files to the **tmp** (temporary) folder on your hard drive. This can be seen in the **Properties Editor, Output Properties, Output tab** (Figure 29.7).

Note: Properties Options Curtailed

Figure 29.7
Figure 29.8

Set the File Path for Saving

You change this setting by clicking on the **Browse Folder** button (Figure 29.3) and navigating to a new folder in the **File Browser Editor.** Select the folder then click on the **Accept button** at the top right hand side. For convenience and simplicity create a new folder. In this demonstration the folder is named **A_Submarine_Movie** and the file path to the folder is: C:\ **A_Submarine_Movie** (Figure 29.8).

Set the Video Format

Set the Movie Video output format (see File Type in Chapter 17 - 17.9 Video Codecs). To demonstrate the movie making process the **AVI Raw** codec (.avi) will be used from the **Movie** list in the **Properties Editor, Output Properties, Output Tab** selection menu (Figure 27.9).

Figure 29.9

Since the **Video Files** (clips) being compiled into a **Movie File** are also **.avi file format** you are, in fact, simply assembling the files into a single file. If you select either the AVI JPEG or FFmpeg video options then the output after assembling would undergo a conversion.

29.8 Video Sequence Editor

VSE Preview superimposed in the **VSE Sequencer**

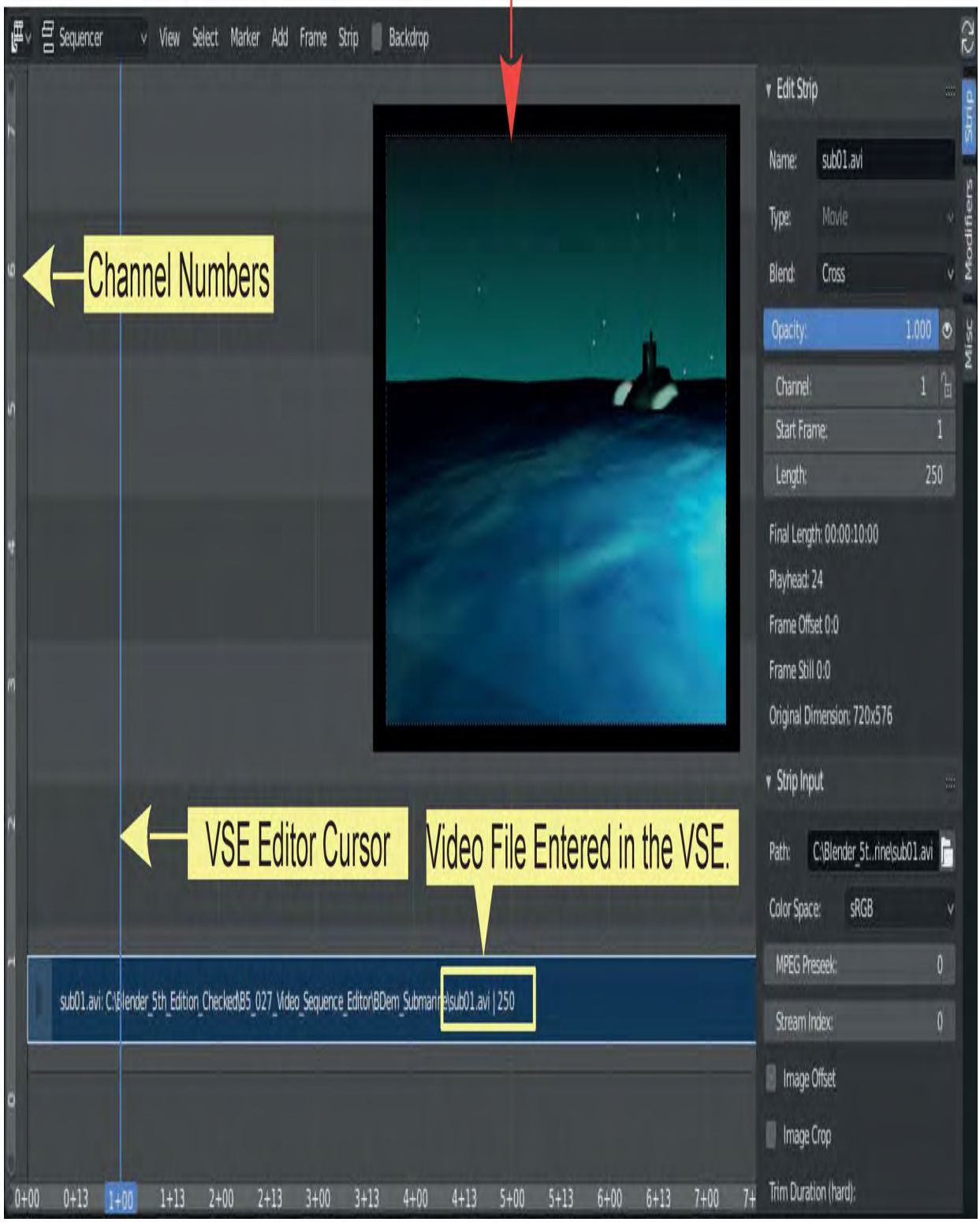


Figure 29.10

The main panel in the **Video Sequence Editor (VSE)** is divided into Channels (horizontal strips), numbered at the left hand side. In Figure 29.10 a Video File named **sub01.avi** has been entered in **Channel 1.** A preview panel has been superimposed in the diagram showing the first Frame of the animation. When a Video File is entered in the Video Sequence Editor the first Frame of the animation displays in the Video Sequencer Preview.

Flacing Fiies in the VSE

To enter a Video File in the VSE, click the file thumbnail in the File Browser, hold and drag into the VSE. By default the first file entered is placed in **Channel 2.**

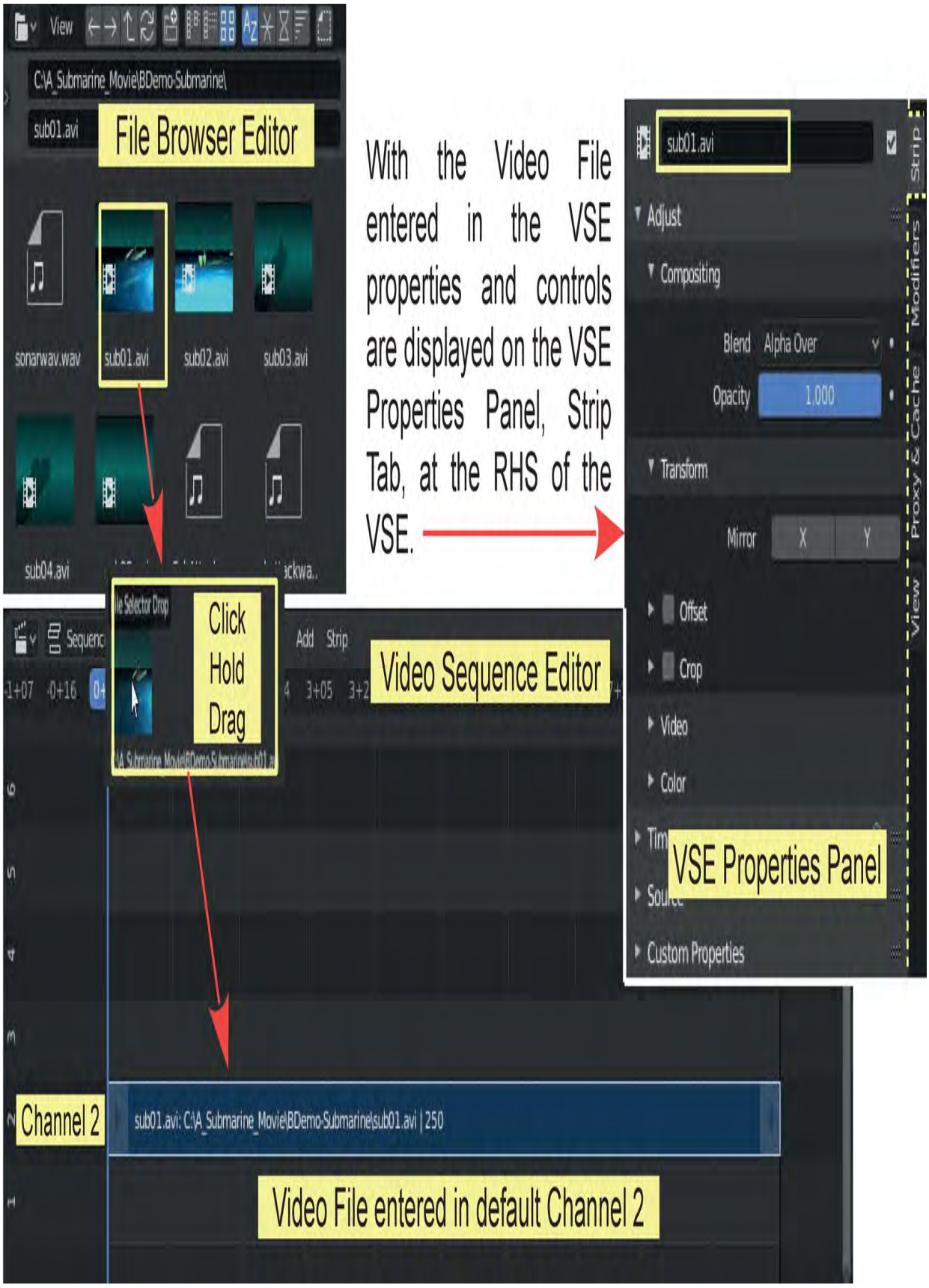

View
C:\A_Submarine_Movie\BDemo-Submarine\
sub01.avi
File Browser Editor
sonarwav.wav
sub01.avi
sub02.avi
sub03.avi
sub04.avi
With the Video File entered in the VSE properties and controls are displayed on the VSE Properties Panel, Strip Tab, at the RHS of the VSE.
sub01.avi
Adjust
Compositing
Blend
Alpha Over
Opacity
1.000
Transform
Mirror
X
Y
Offset
Crop
Video
Color
Custom Properties
Strip
Modifiers
Proxy & Cache
View
Click
Hold
Drag
Add
Strip
Video Sequence Editor
VSE Properties Panel
Channel 2
sub01.avi: C:\A_Submarine_Movie\BDemo-Submarine\sub01.avi | 250
Video File entered in default Channel 2

Figure 29.11

Placing Files in the VSE (Alternative Method)

Various types of files may be entered in the VSE and combined with video files. Click on **Add** in the VSE Header (Figure 29.12) and select what you wish to enter (the File Browser Editor opens). In this instance you are entering a Video File, therefore, select **Movie.** Navigate to the folder containing your Video Files and select a file. Click Add Movie Strip (upper RH corner of Screen). This is an alternative to the method previously described. By default the File is entered in **Channel 2.** Note: By default the VSE Editor Cursor is located at position 0+01 in the **Playback Timeline** along the bottom of the VSE. When Files are entered in the VSE they are located at the position of the Cursor.

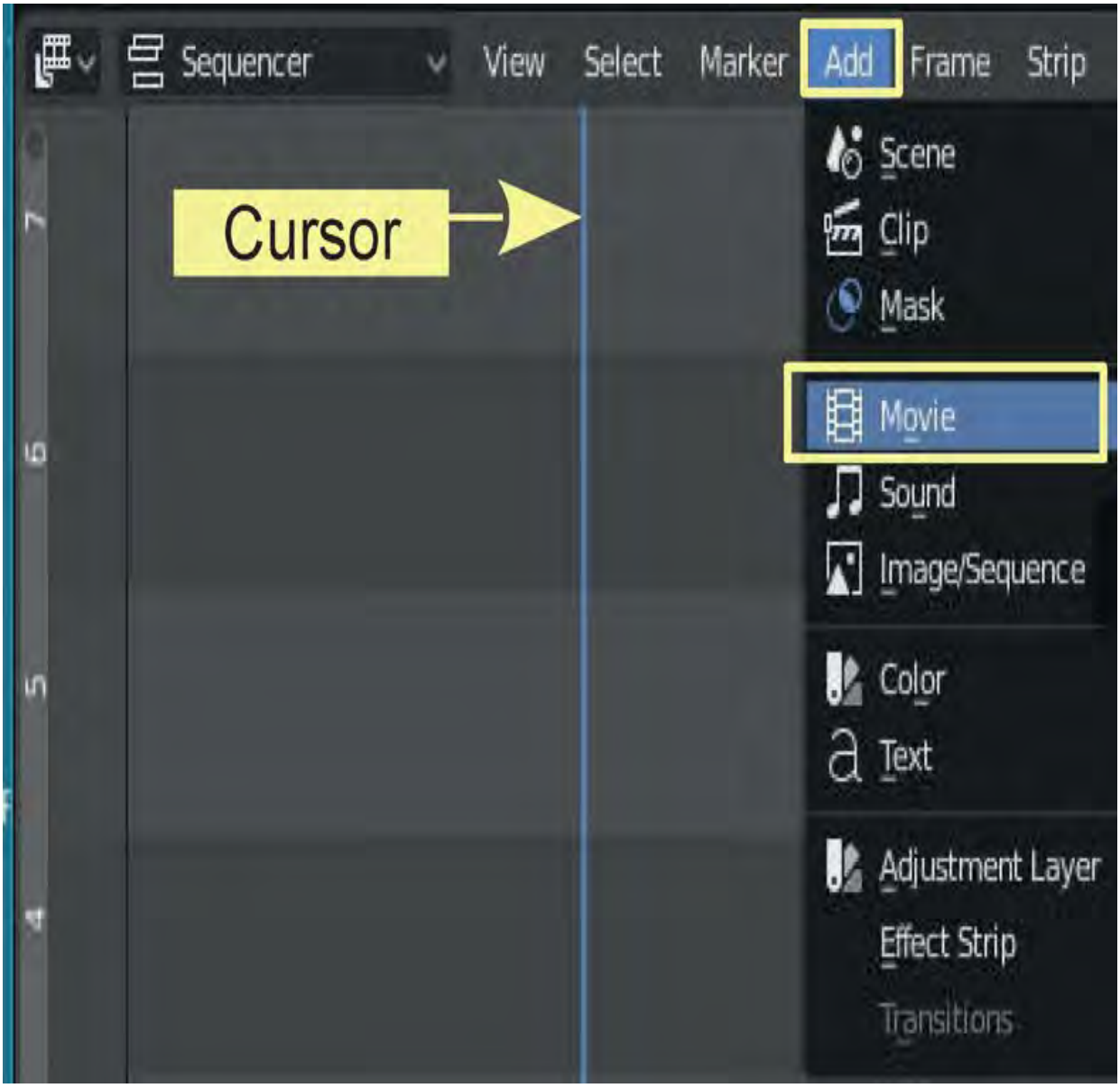


Figure 29.12

Viewing the Video File

To see Video Files in action click the Play button in the Timeline Editor at the bottom of the Screen. You may also click, hold and drag the VSE Cursor to scrub through the Video Files.

Video Files in upper Channels take precedence and play over lower Channels.

Selecting in the VSE

You select a File in a Channel by clicking LMB. Hold and drag R or L to reposition. LMB click on a file, hold and drag up or down to place the file in a different Channel.

Note: When repositioning horizontally, click LMB, hold and drag then you may release MB. When positioning horizontally you will see a Frame Number appear at the beginning and end of the Video File giving you the exact location in the Timeline.

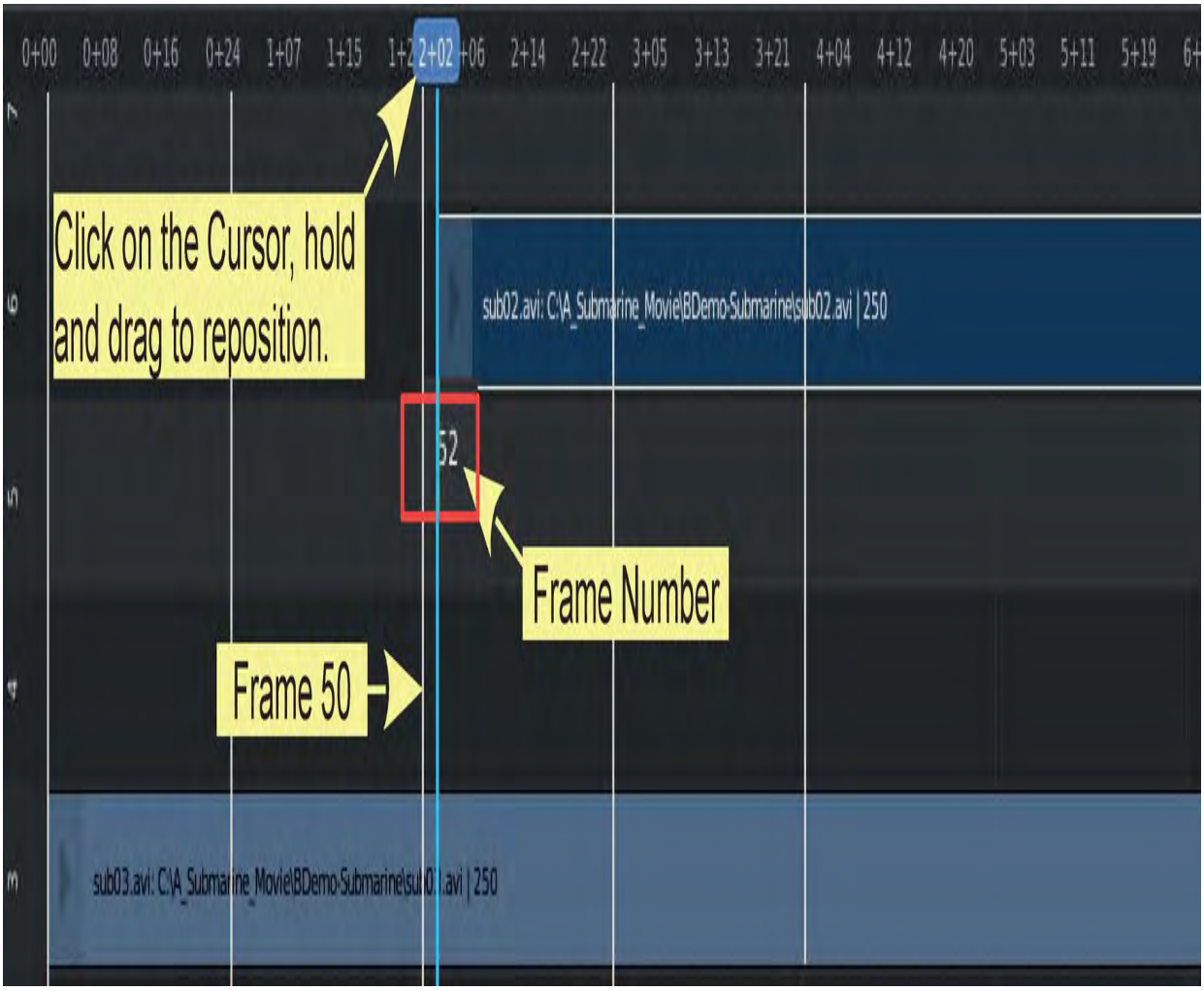


Figure 29.13

Note: The Cursor is positioned at Frame 2 + 02 (see Timeline Graduations / Positions following.

Timeline Graduations / Position

Example: File named sub02.avi is positioned at Frame 52. In the Timeline the position is given as 2+02. The horizontal divisions (faint vertical lines) are located at:

0+00 = Frame 0.00 1+00 = Frame 25 2+00 = Frame 50

(25 Frame Divisions)

Frame 52, is therefore 2 + 02
(2 × 25) + 2 = 52

Erasing (Deleting) a File

LMB click on the File in the VSE (border highlights white), press the **X Key** or press Delete or RMB click and select delete.

The Add Button

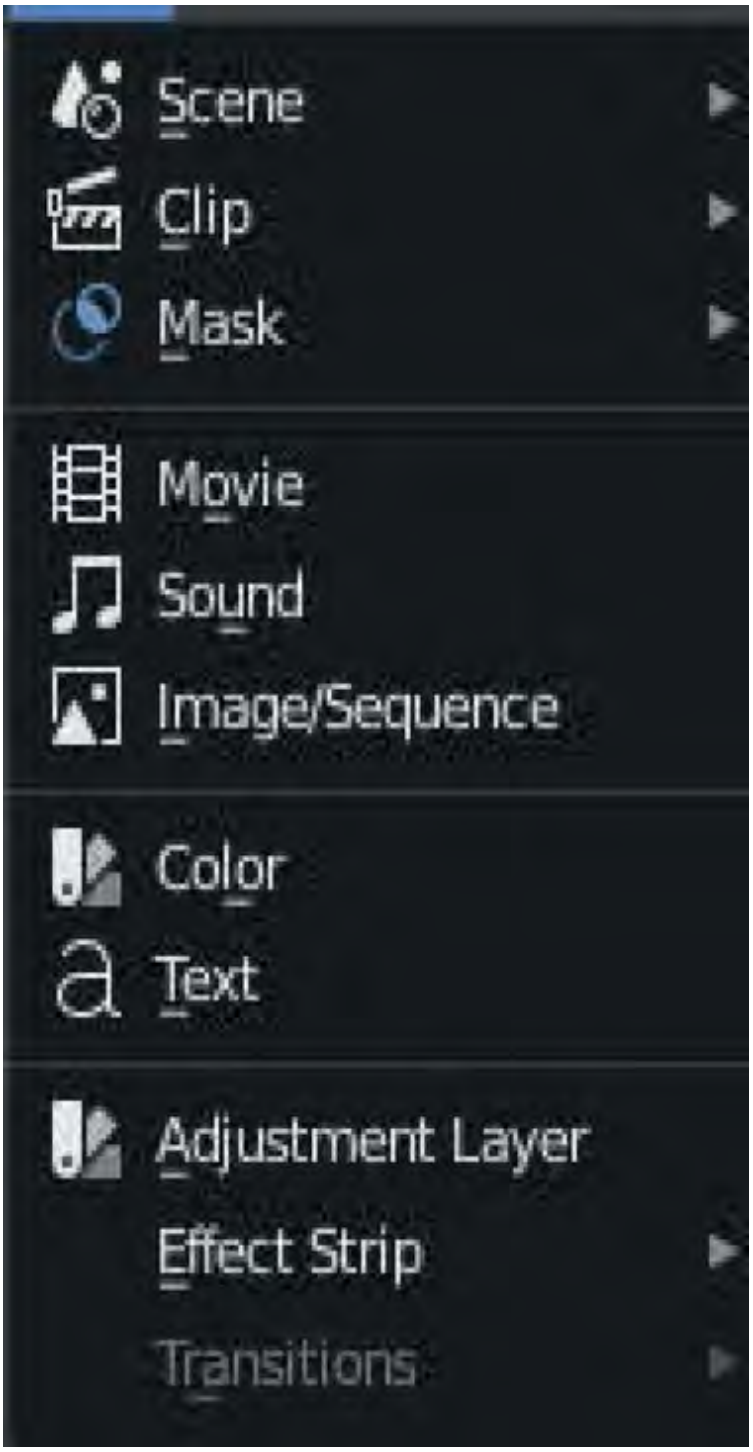


Figure 29.14

The **Add button** in the **VSE Panel** header has several options.

Scene: Adds a strip containing information about a Scene in the Blender file.

Mask: If a mask has been created it can be added to the VSE to hide or alter the appearance of parts of the video.

Image: A still image or a series of images may be inserted into the video much like adding individual frames of an animation or a slide show.

Sound: Sound files can be inserted in the VSE to enhance video.

Effects Strip Effects to provide enhancement, background and transition

With the forgoing information you are in a position to proceed and compile the Movie File.

Adding Video Files

When adding Video Files it is helpful to scale and pan the VSE Editor. This allows you to get a bigger picture of your assembly. With the Mouse Cursor in the VSE Editor you can zoom in and out by pressing the Plus and Minus keys on the keyboard or by scrolling MMB.

At the bottom of the VSE Sequencer is a gray bar with dots at each end. Lick on the bar, hold and drag left or right to pan the display in the VSE. Click hold and drag the dots at either end to scale the VSE view horizontally. A similar vertical pan and scale bar is at the RHS of the VSE.

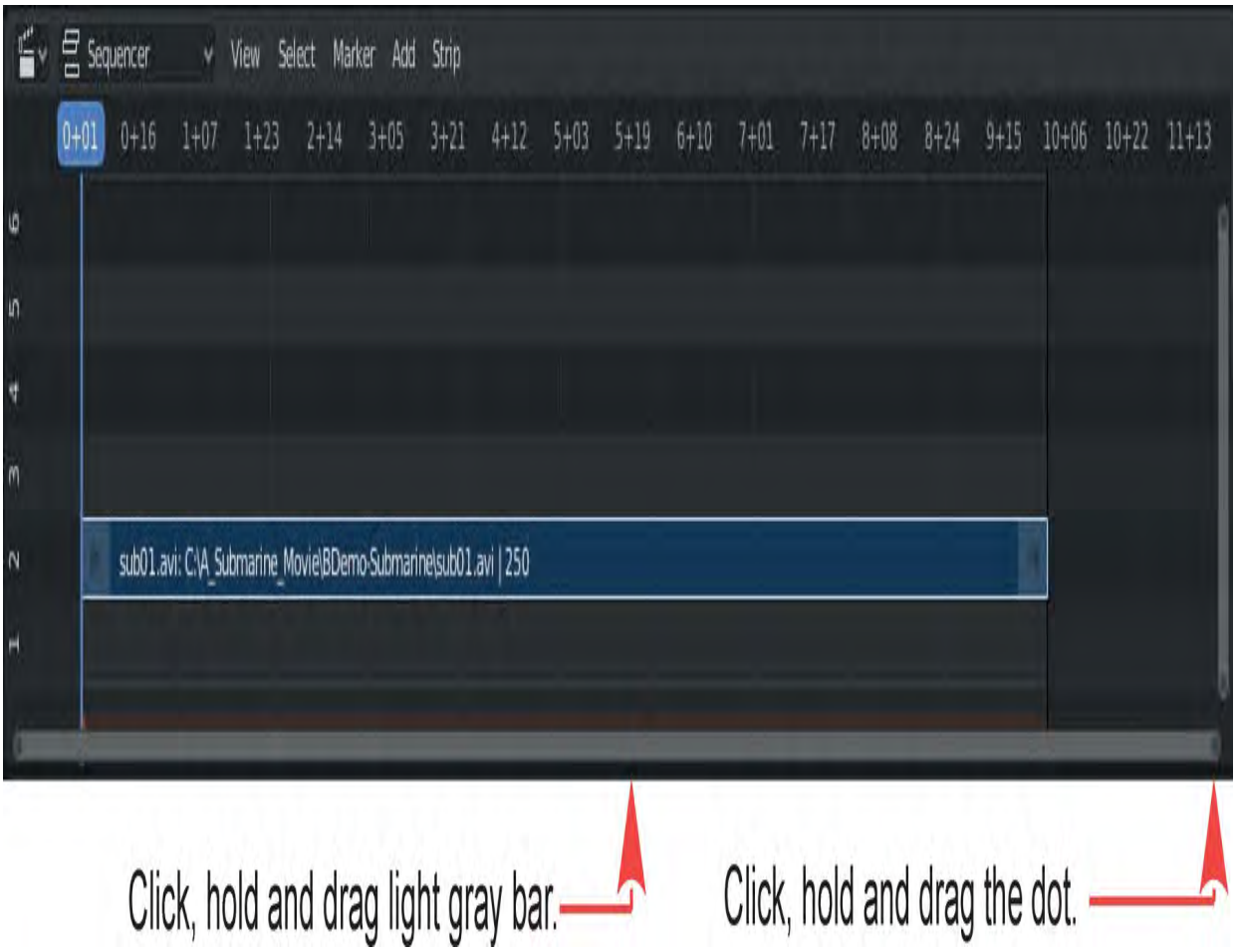


Figure 29.15

As you have seen you add Movie Files by clicking, holding and dragging from the File Browser Editor or clicking **Add** in the **VSE Editor header** (Press Add — Movie — navigate in the **File Browser window** — select etc.).

The first file is entered by default in Channel 2 with the VSE cursor located at 0+01. To add a second file, position the cursor where you want it to start and repeat the Add process. A second file is entered in Channel 3. The files can be moved to different channels as you wish and repositioned horizontally.

Note: If the **VSE Cursor** is placed at the end of the first Movie File the second file will be placed in the same Channel as the first, end to end.

Each successive file addition is entered in the Channel above the preceding Channel except when the VSE Cursor is positioned at the end of the preceding file. To have two Movie Files play end to end as a continuous sequence, position the start of the second file horizontally at the end of the first file (they do not have to be in the same Channel). With the second file selected, press the **G Key** and drag the Mouse. You will see Frame Numbers display at the beginning and end of each file which makes it easy to align exact Frames. You can purposely overlap files since a file in a higher Channel will take precedence over a file in a lower Channel when playing.

Playing the Video File

No matter where the file is located you can view different Frames in the file by dragging the Cursor along the Timeline. You play the file by pressing the **Start button** in the **Timeline Editor.** Press **Esc** to quit or Pause, Fast Forward etc. by using the play controls.

To mention some more obvious information about playing consider this; the Video Files used in the demonstration are 250 Frames long. With the first located with the **Start Frame** at 0+01 it will play in its entirety then repeat until you press **Esc.** This only occurs since, in the **Timeline Editor, Start: 1** and **End: 250** are set. If **End: 100** was set the file would only play for 100 Frames then repeat, or if the start Frame of the file was positioned on the VSE Timeline other than at Frame 1 then only part of the file will play. (see Figure 29.16) over.

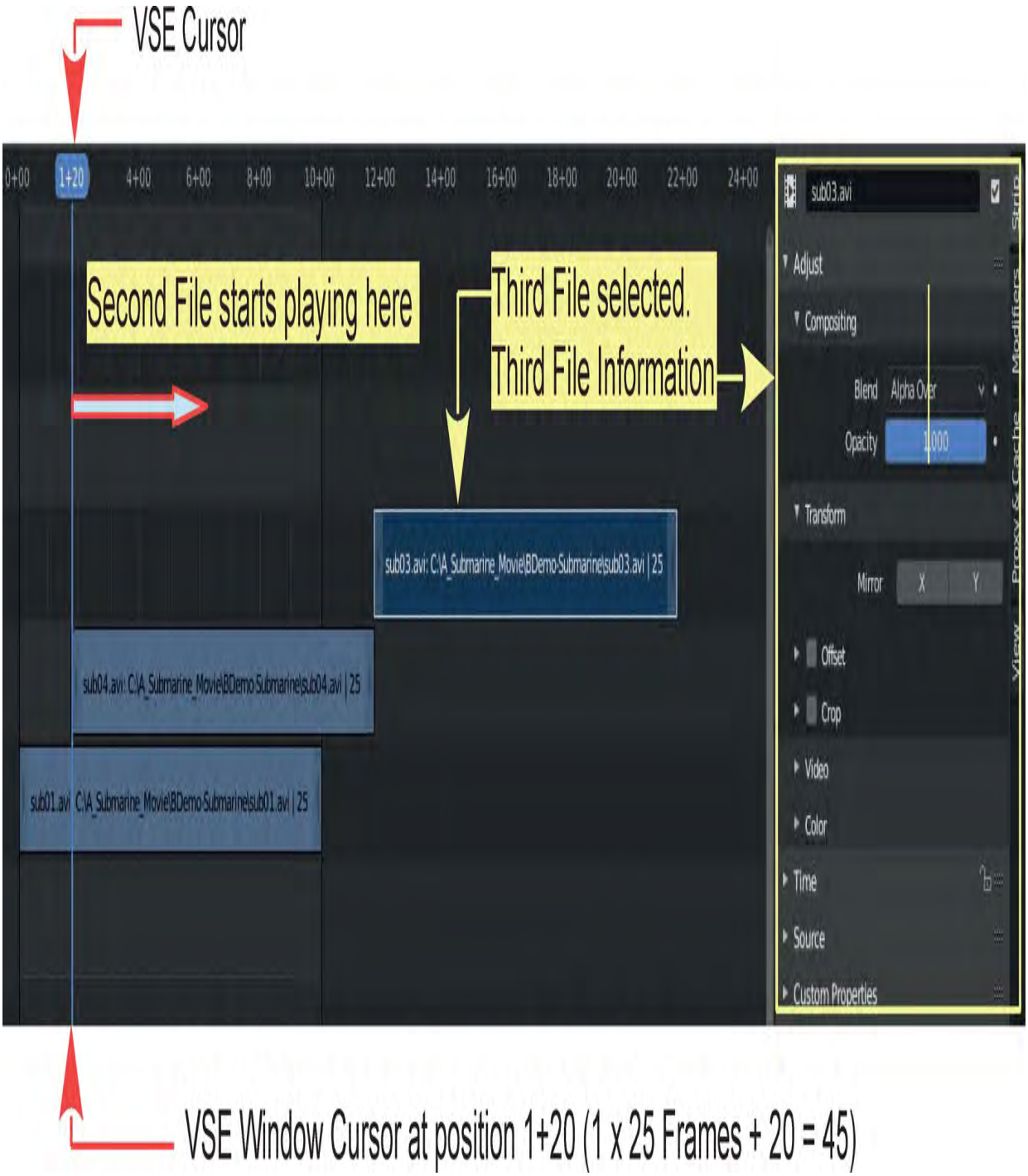


Figure 29.16

Cutting Video Strips

A Video File in the VSE is also referred to as a **Video Strip.** Another feature of the VSE is the ability to select only part of a video strip for playback. You

can cut the strip into segments. There are two ways to do this which are; a **Soft Cut** and a **Hard Cut.** In either case, position the Cursor at the Frame where you wish to make the cut. For a **Soft Cut** press the **K Key.**

For a **Hard Cut** press **Shift + K key.** In either case you finish up with two separate segments of strip which you can reposition or move to a different Channel in the VSE. The difference is, with a **Soft Cut** both segments of the strip retain the data for the other part. With a **Hard Cut** the data is not retained (Figure 29.17).

Figure 29.17

Note: When Cut, File: sub01.avi divides into subi .ave + sub001.avi

Adding Sound File'

Sound files such as MP3 and WAVE are entered by selecting **Sound** in the **Add button menu** instead of **Movie** and then manipulating the same as a Video File.

With all your strips aligned and edited you can press the **Play button** in the **Timeline Editor** to preview the final movie.

29.9 Rendering the Movie Fil€

When all the specifications have been set for your Movie Output File it is time to render the final movie.

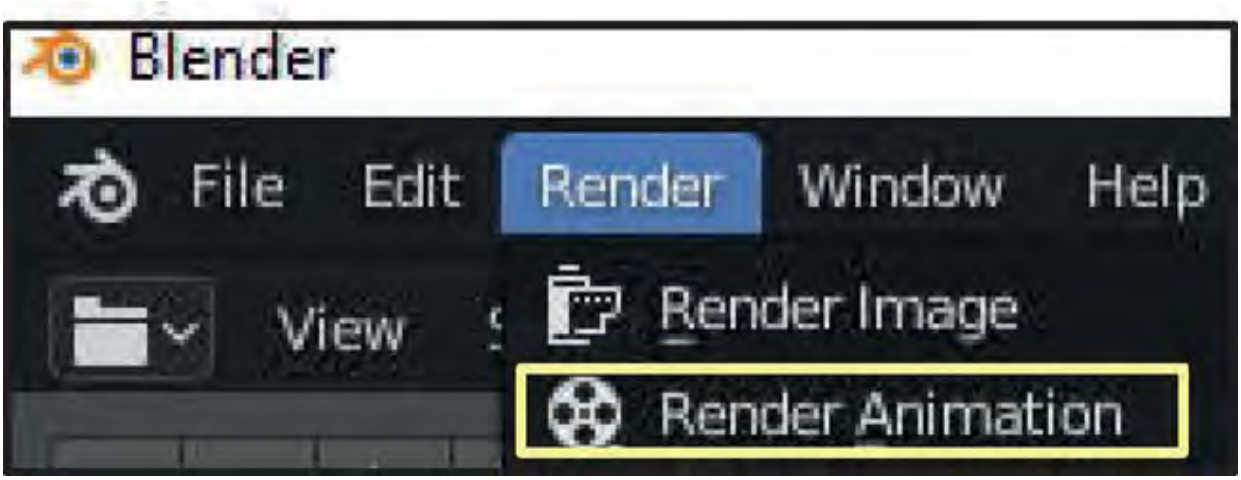


Figure 29.18

In the **Screen Header** click the **Render button** and select **Render Animation.** Be prepared to wait a considerable time. Even a short movie will take awhile depending on the speed of your computer. Long movie sequences are often uploaded to websites called **Render Farms** which will perform the render process for you (at a cost). Once the render is complete you find the file in the output folder and give it a test run in a media player.

29.10 Additional Features

Additional features for enhancing and modifying video strips are accomplished by adding features from the VSE Header Add button menu or by the application of Strip Modifiers in the VSE Properties Panel.

Generally adding a feature from the Header inserts a special Feature Strip in the VSE in addition to the Video Files entered, while the application of a Modifier affects the selected Video Strip in the VSE.

sub01.avi
Strip
Adjust
Compositing
Blend
Alpha Over
Opacity
1.000
Modifiers
Transform
Mirror
X
Y
Proxy & Cache
Offset
Crop
View
Video
Color
Time
Source
VSE Properties
Custom Properties

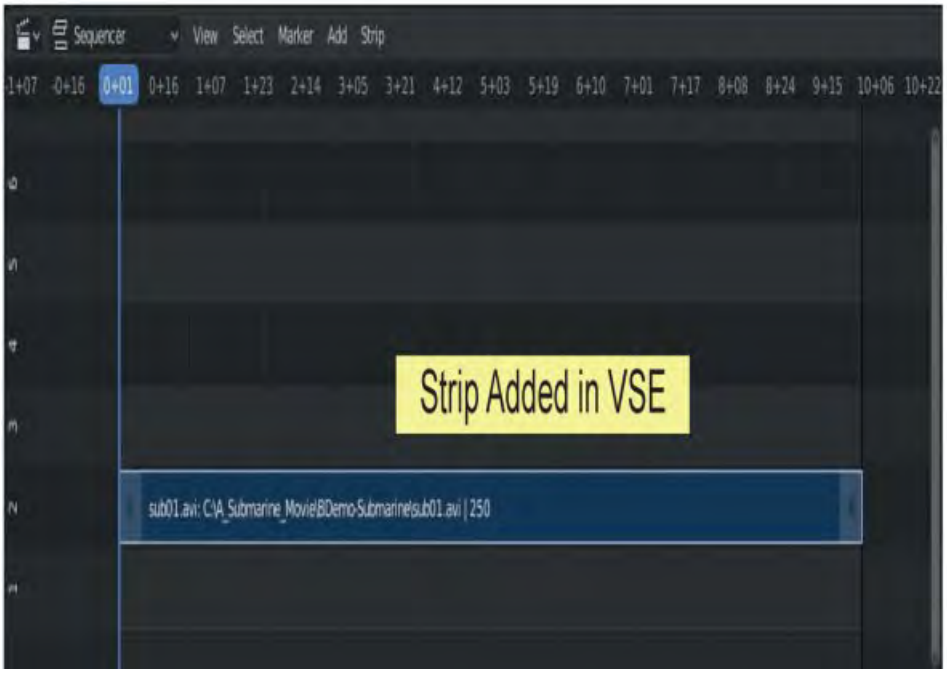

Sequencer
View
Select
Marker
Add
Strip
1+07
0+16
0+01
0+16
1+07
1+23
2+14
3+05
3+21
4+12
5+03
5+19
6+10
7+01
7+17
8+08
8+24
9+15
10+06
10+22
Strip Added in VSE
sub01.avi: C:\A_Submarine_Movie\BDemo-Submarine\sub01.avi | 250

Figure 29.19

Example 1: No Features Applied

Preview with Strip in VSE and default settings in the VSE Properties. **No additional features.**

Example 2: Add Color Feature

Sequencer
View Select Marker Add Strip
Click Add
Color Strip Added
Scene
Clip
Mask
Movie
Sound
Image/Sequence
Color
Text
Adjustment Layer
Effect Strip
Transitions
sub01.avi C:\A_Submarine_Movie\BDemo\Submarine\sub01.avi | 250
Color is applied in the preview for the length of the Color Strip
Color
Adjust
Compositing
Blend Alpha Over
Opacity 0.409
Transform
Mirror X Y
Offset
Crop
Video
Color
Effect Strip
Color
VSE Properties
Strip
Modifiers
Proxy & Cache
View

Figure 29.20

Click **Add** in the VSE Header and select **Color** in the menu. A Color Strip is added in the VSE Editor and controls display in the VSE Properties Panel in the Strip Tab.

Select a Color and reduce **Opacity**

Example 3: Apply Color Modifier

Modifiers
Use Linear Modifiers
Add Strip Modifier
Copy to Selected Strips
Strip
Modifiers

Modifiers
Use Linear Modifiers
Add Strip Modifier
Copy to Selected Strips
Color..
Mask Input Type
Strip
Mask
Mask
Multiply Colors
1.000
Lift:
Invert
Gamma:
Invert
Gain:
Invert
Strip
Modifiers
Proxy & Cache
View

Figure 29.21

Select a Strip in the VSE. In the VSE Properties, Modifiers Tab click Add Strip Modifier. In this case select Color. Alter the values in the tri color pickers to adjust the preview display. Color is applied permanently for the whole Strip.

Note: The Strip must be selected in the VSE.

Example 4: Inserting Text Captions

Examples 2 and 3 were concerned with affecting the Preview Color Tones.

Captions may be inserted in a Strip to display a Title, credits etc.

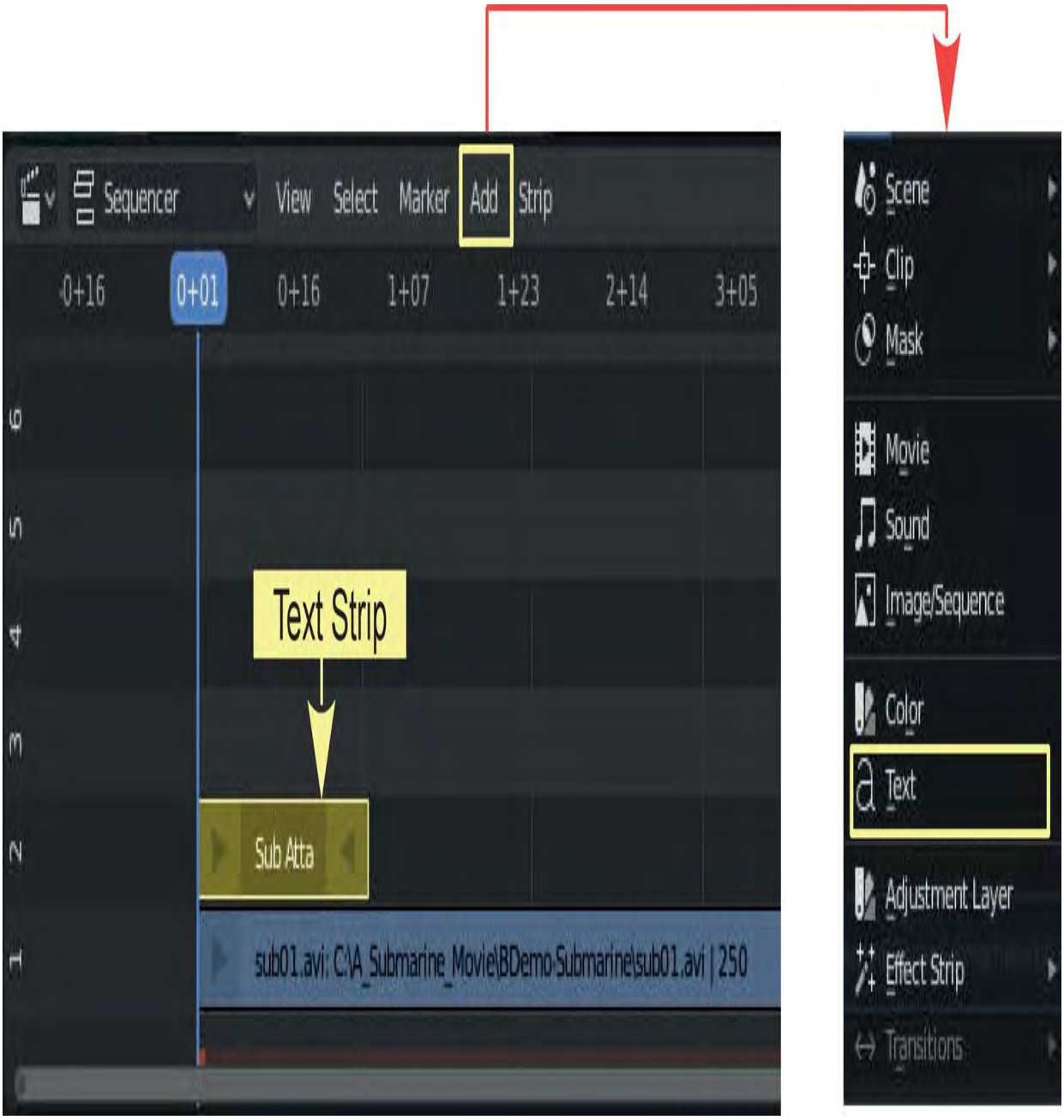


Figure 29.22

Select a Strip in the VSE. In the VSE Sequencer Header click on **Add** and select **Text** from the menu that displays.

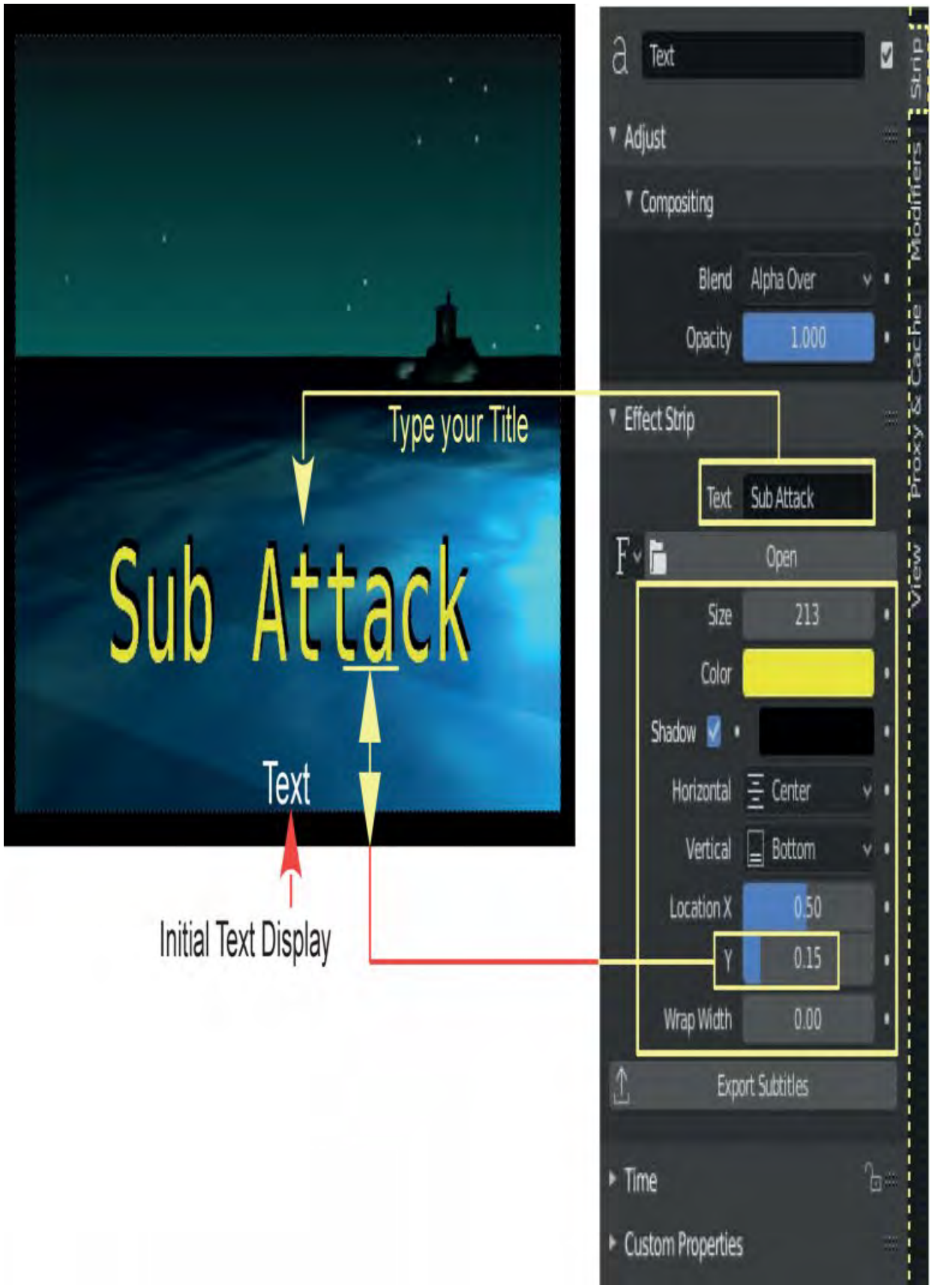

Type your Title
Sub Attack
Text
Initial Text Display
Text
Adjust
Compositing
Blend
Alpha Over
Opacity
1.000
Effect Strip
Text
Sub Attack
Open
Size
213
Color
Shadow
Horizontal
Center
Vertical
Bottom
Location X
0.50
Y
0.15
Wrap Width
0.00
Export Subtitles
Time
Custom Properties
View
Proxy & Cache
Modifiers
Strip

Figure 29.23

Selecting **Text** in the menu inserts the small white word **Text** at the bottom of the video display in the VSE Preview. A **Text Strip** is added in a Strip at the **position of the VSE Cursor.** In the **VSE Properties Panel** at the RHS of the VSE is the **Strip Tab** with controls for modifying the Text.

Experiment with the controls. I what appears to be a simple a simple arrangement of controls will produce a variety of effects.

30 Drivers

30.1 Blender Drivers Introduction
30.2 The Driver Editor
30.3 Understanding Drivers
30.4 Randomize Object Properties
30.5 Using Random.Py.blend
30.6 Material Property Drivers
30.7 Duplicating the Object

Drivers are functions or scripts which use properties (values) to affect other properties. This is particularly applicable when animating. Drivers are used to control the animation of one property based on the value of another.

For example, the Translation (movement) of one Object may be used to control the Rotation of another Object. This means that the Object's animated value is not controlled by the Frame number interpolated from Keyframes, but rather by the data in a specified animation channel. Drivers can take their effects from single properties, differences in rotation or scripted Python expressions which may be edited inside the User Interface controls.

Drivers are not limited to simple animated movements. You may use the X location of a Driver of an Object to control the Material color (RGB curves) of another Object's Material, or use the rotation of a Driver to control the Scale of an Object, or use the Scale of a Driver to control the shape (through Shape Keys) of a Mesh/Curve/etc., use a Python function to control a constraint's influence, and much much more.

One key usage of Drivers is in Character Animation: for example, you can add Object Drivers to the relative Shape Keys of a Face. Then, you manipulate the expressions of your character just by moving these Drivers' Objects.

To explain Drivers in detail would require a dedicated publication. This chapter will simply demonstrate, in practical terms, what a driver is and where the controls are located in the interface. This will provide a starting point for a detailed study.

30.1 Blender Drivers – Introduction

As a simple example of using **Drivers,** a Monkey Object will Rotate on its Z Axis when a Cube Object in the Scene Translates on the X Axis.

Open Blender with the default Scene containing the Cube object. Deselect the Cube and add a Monkey Object. Move the Monkey to one side as shown in Figure 30.1.

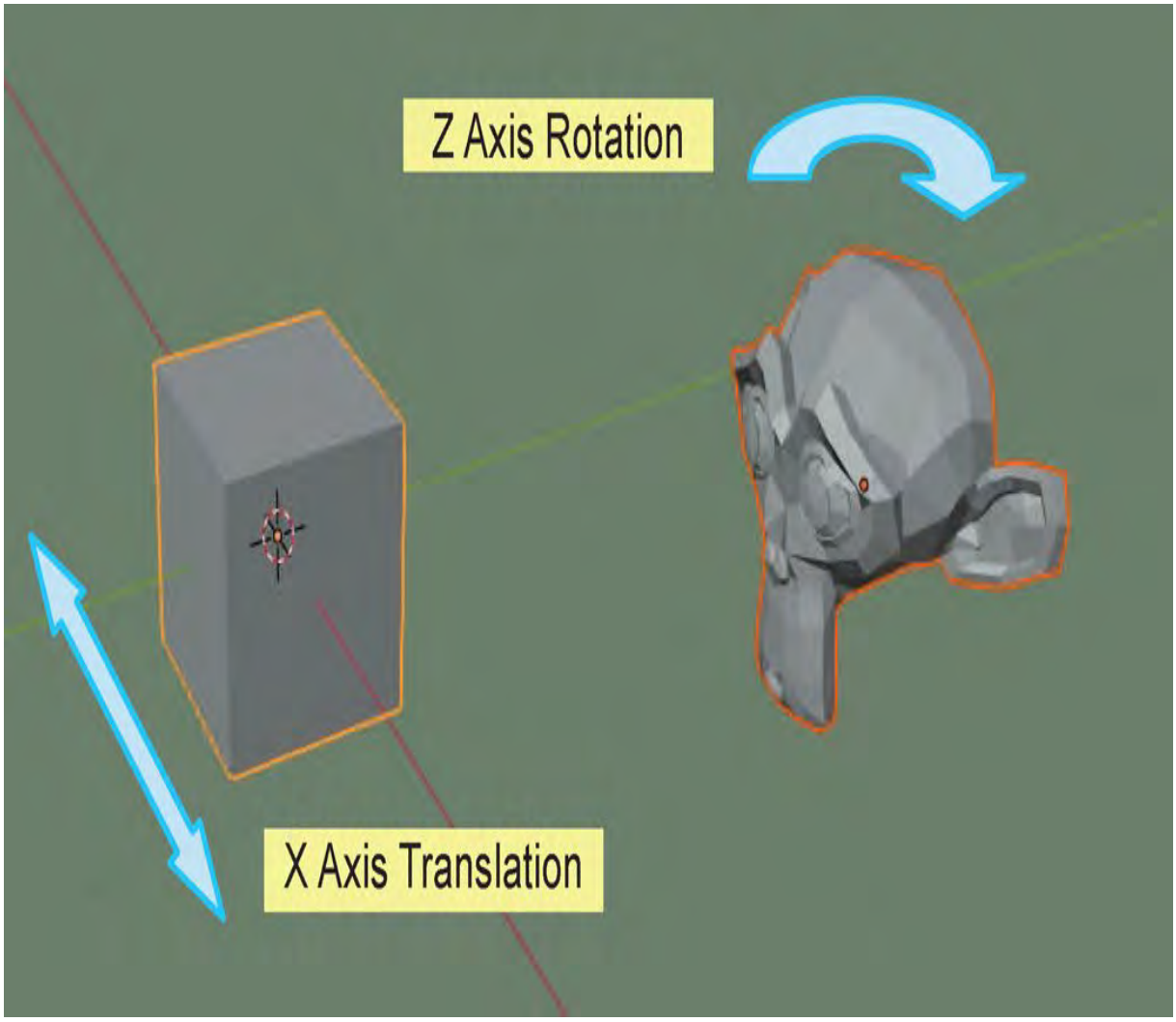


Figure 30.1

With the Monkey Object selected and with the Mouse Cursor in the 3D Viewport, press the **N Key** to display the **Object Properties Panel** for the Monkey

RMB click on **Z Axis Rotation** and select **Add Driver** in the menu that displays. The Z Rotation value bar turns purple and the **Driven Property Panel** opens.

Note: Type: X Location.

Figure 30.2

In the **Driver Property** panel change **Type: Scripted Expression** to **Maximum Value** and click on **Object** and select Cube.

You have set the Driver for the Monkey to be controlled by the **X Axis** movement of the Cube. In the 3D Viewport Editor select the Cube and Translate it along the X Axis. The Monkey Rotates on its Z Axis (Figure 30.1).

In setting up the driver the **Driver Properties Panel** has disappeared from view. Should you wish to Edit the Properties, select the Monkey in the Viewport Editor, press the N Key to open the Object Properties Panel. RMB click on the purple Z Axis Rotation value. Select **Edit Driver**. The **Driven Properties Panel** is reinstated allowing a limited Edit. For a comprehensive Edit, open the **Drivers Editor**.

There are two ways to open the **Drivers Editor**. Divide the 3D Viewport Editor into two panels and make one panel the Drivers Editor, alternatively in the Object Properties Panel, Transform Tab, RMB Click on Z Axis Rotation (purple bar) and select Open Drivers Editor. In the later, the Drivers Editor is opened as an overlay in the 3D Viewport Editor.

30.2 The Drivers Editor

The **Z Euler Rotation Driver** only displays when Suzanne is selected in the 3D Viewport.

To display **Driver Property** press the N Key with the Mouse Cursor in the Driver Editor. Have the **Drivers Tab** selected.

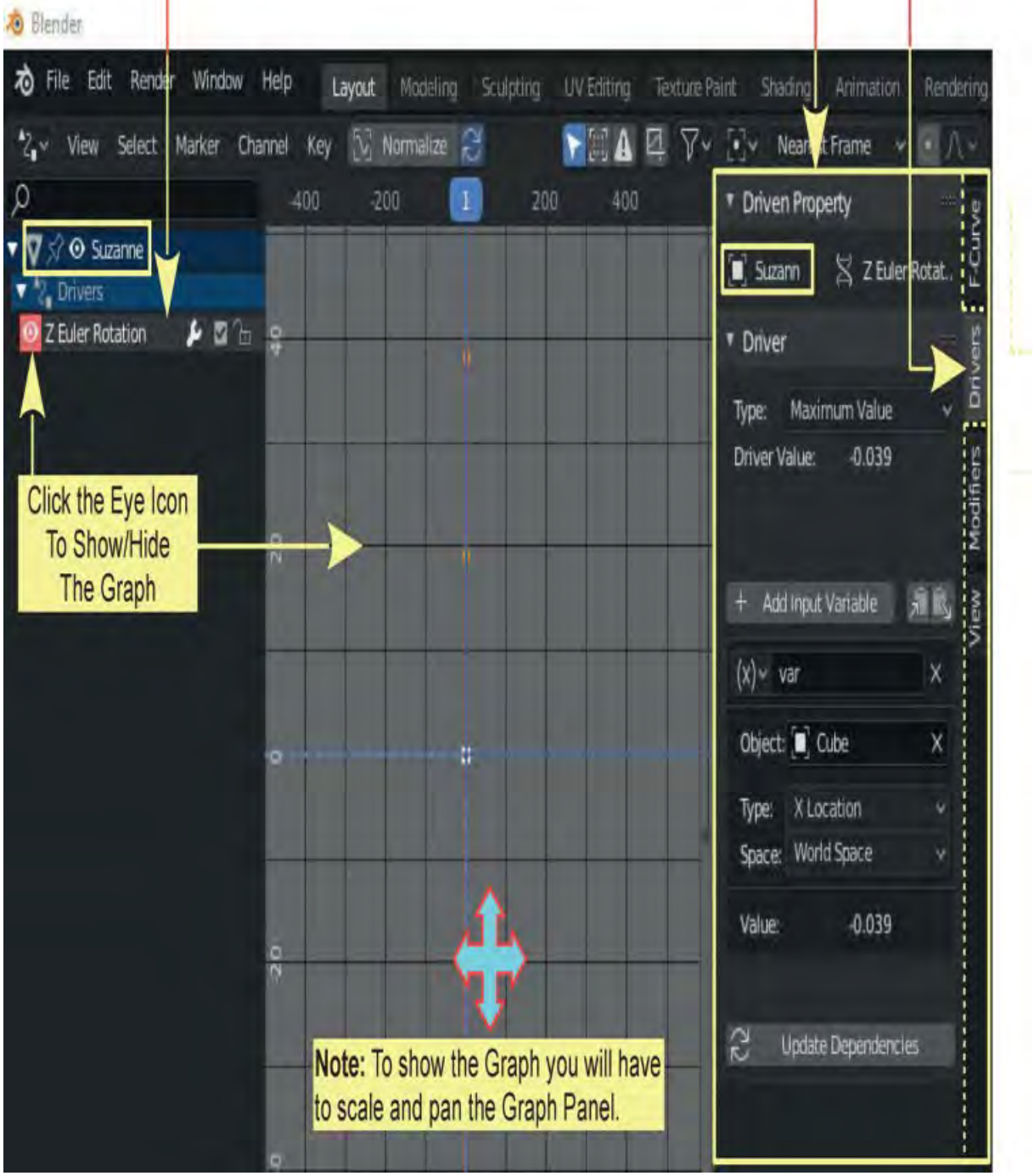


Figure 30.3

Important: The view of the **Driver Editor** shown in Figure 30.3 is the Editor as displayed with the **Susanne** (Monkey) **selected** in the 3D Viewport and with the **Z Rotation Driver in place**. You see the Driven Property Panel, previously shown in the 3D View Editor when adding the driver, is replicated at the RHS of the Driver Editor. **This is when the Drivers Tab is selected**.

30.3 Understanding Drivers

The example previously described explains the basic concept of applying Drivers but understanding Drivers in relation to Blender's interface will allow you to use them for a variety of applications. Look at the Object Properties, Panel, **Transform Tab** in the 3D Viewport Editor (N Key to display) and the **Driven Property** in the **Driver Editor**.

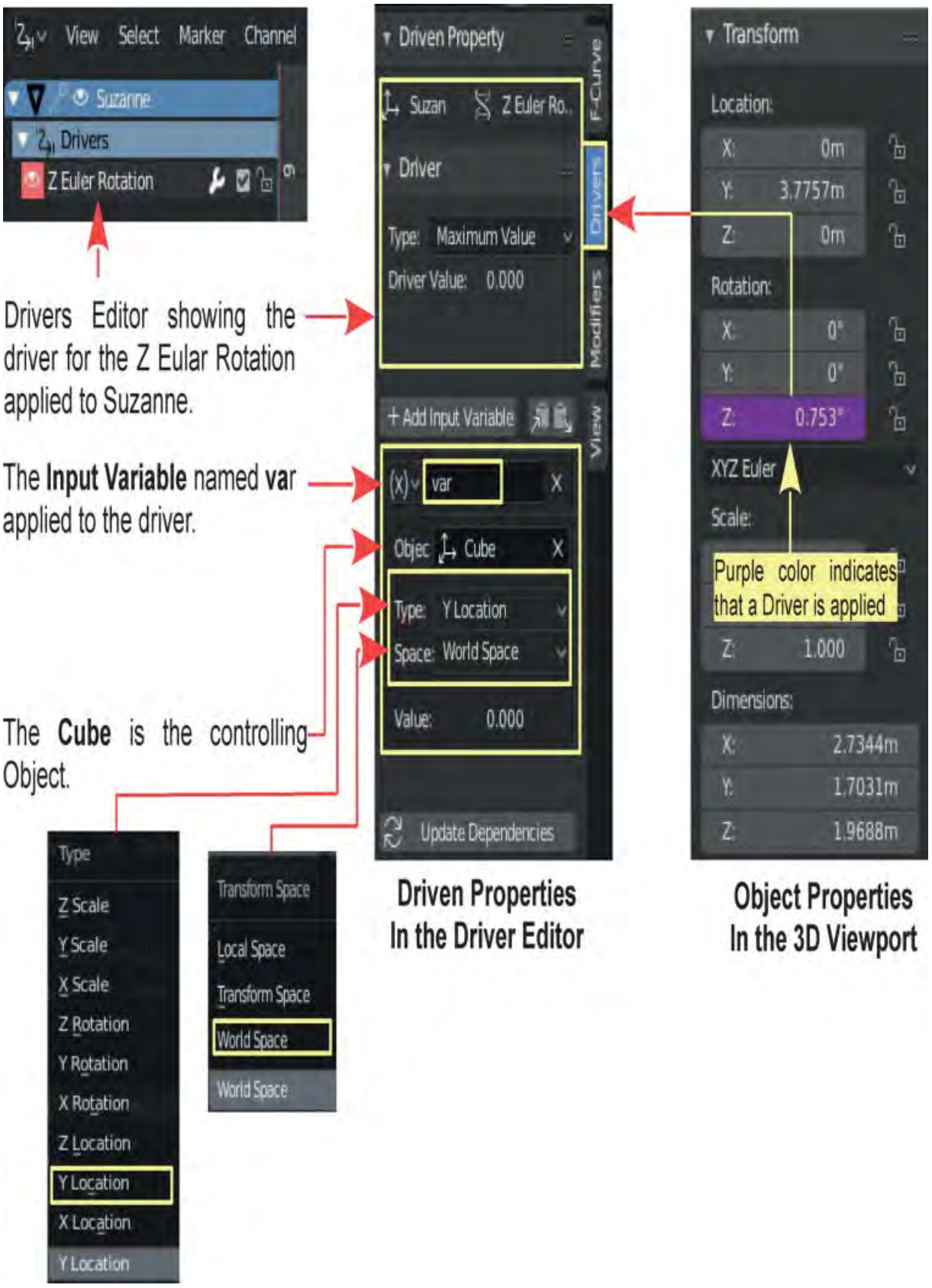

View Select Marker Channel
Suzanne
Drivers
Z Euler Rotation
Drivers Editor showing the driver for the Z Eular Rotation applied to Suzanne.
The **Input Variable** named **var** applied to the driver.
The **Cube** is the controlling Object.
Driven Property
Suzan
Z Euler Ro..
Driver
Type: Maximum Value
Driver Value: 0.000
F-Curve
Drivers
Modifiers
View
+ Add Input Variable
(x) var
Objec: Cube
Type: Y Location
Space: World Space
Value: 0.000
Update Dependencies
Type
Z Scale
Y Scale
X Scale
Z Rotation
Y Rotation
X Rotation
Z Location
Y Location
X Location
Y Location
Transform Space
Local Space
Transform Space
World Space
World Space
Transform
Location:
X: 0m
Y: 3.7757m
Z: 0m
Rotation:
X: 0°
Y: 0°
Z: 0.753°
XYZ Euler
Scale:
Purple color indicates that a Driver is applied
Z: 1.000
Dimensions:
X: 2.7344m
Y: 1.7031m
Z: 1.9688m
Driven Properties In the Driver Editor
Object Properties In the 3D Viewport

Figure 30.4

Space: (Transform Space):World Space

Type: Y Location (the Y Axis location of the Cube determines the Rotation of Suzanne).

Type and **Transform Space** values entered in this particular example.

Take a step back and in a new Blender file with the **default Cube selected** add a Driver to the **X Axis Transform Location** slider. The slider displays purple. **Click RMB on the purple slider** and select **Open Drivers Editor** in the menu to open the **Drivers Editor Panel.**

The Drivers Editor (Panel) displays with the X Location Driver active and a straight line graph for the X Location of the Cube. The Driven Property panel is also displayed where you may edit values to control the Location of the Cube in the 3D Viewport Editor.

Values in Driven Property Panel

Delete Expression: var+0.0 and enter 2 (Figure 30.5). The Cube moves + 2 Units on the X Axis in the 3D Viewport Editor.

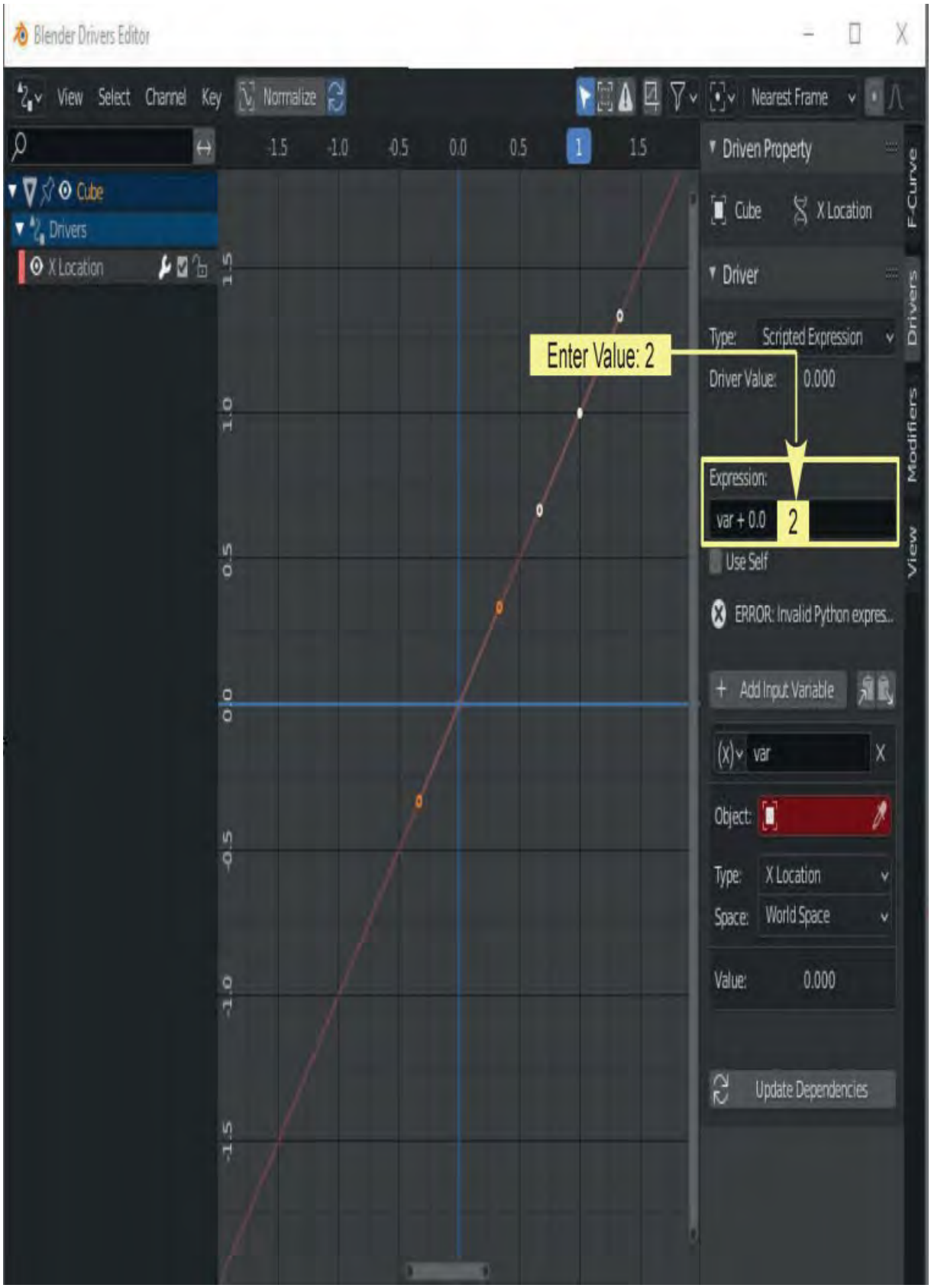

Blender Drivers Editor
View
Select
Channel
Key
Normalize
Nearest Frame
Cube
Drivers
X Location
-1.5
-1.0
-0.5
0.0
0.5
1
1.5
Driven Property
Cube
X Location
Driver
Type:
Scripted Expression
Driver Value:
0.000
Enter Value: 2
Expression:
var + 0.0
2
Use Self
ERROR: Invalid Python expres...
Add Input Variable
(x)
var
X
Object:
Type:
X Location
Space:
World Space
Value:
0.000
Update Dependencies
F-Curve
Drivers
Modifiers
View

Figure 30.5

Instead of 2 enter 2*2 (2 x 2) .The Cube moves 4 Units'

Entering 3*4/6 (3 x 4 ÷ 6) the Cube locates at + 2 Units.

This is demonstrating that maths can be performed in the Expression panel to set values for the Driver.

Multiple Drivers

You may have Drivers for any or all of the Transform Channels in the Object Properties Panel as well as many of the Properties in the Properties Editor.

Moving Forward

The use of Drivers for controlling Animation is extensive and requires a detailed study beyond the scope of this publication but to encourage the pursuit of this topic the following exercise is offered.

30.4 Randomize Object Properties

This exercise will demonstrate how to set up the **Blender file** including a **Python Script** which may be used with **Drivers**. Once the Blender File is created it may be saved to use at a later time when you wish to randomly control the properties of an Object.

Drivers

In simple terms **Drivers** are functions which affect the attributes or properties of an Object. Refer to the demonstration at the beginning of this chapter where the translation of one Object controls the rotation of another. In that instance the position of one Object in the Scene controlled the rotation of the other. Instead of using the translation of an Object, the **Python Script** will be introduced to the Driver to control properties of Objects.

Blender is written in the Python computer Language. A **Python Script** is a piece of code written in the **Python Computer Language** and is simply a text

file. This text file may be used in any new Blender file but having a saved Blender File with the Python Script installed saves you time in setting up.

Create a Python Script

In a new Blender File change the 3D Viewport Editor to the **Text Editor** (Figure 30.6).

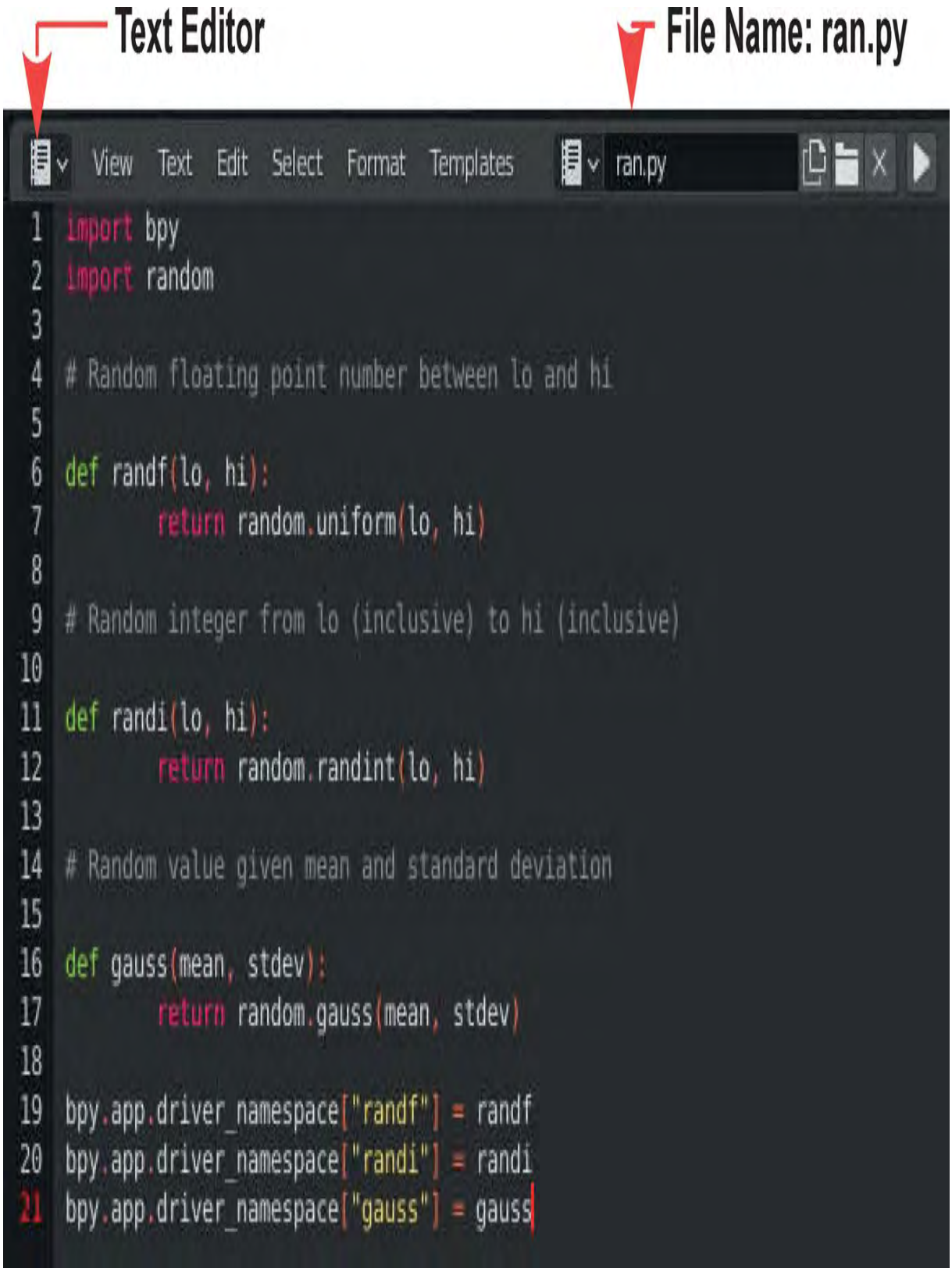

Text Editor
File Name: ran.py
View Text Edit Select Format Templates
ran.py
1 import bpy
2 import random
3
4 # Random floating point number between lo and hi
5
6 def randf(lo, hi):
7 return random.uniform(lo, hi)
8
9 # Random integer from lo (inclusive) to hi (inclusive)
10
11 def randi(lo, hi):
12 return random.randint(lo, hi)
13
14 # Random value given mean and standard deviation
15
16 def gauss(mean, stdev):
17 return random.gauss(mean, stdev)
18
19 bpy.app.driver_namespace["randf"] = randf
20 bpy.app.driver_namespace["randi"] = randi
21 bpy.app.driver_namespace["gauss"] = gauss

Figure 30.6

Type the text shown in the diagram into a Text Editor. Blenders Text Editor enters syntax highlights (the colored text) as you type which is designed to help when corrections are required.

You could type the text into an application such as Notepad or Wordpad and save the text file but to create the Blender File you will have to copy and paste the Text into Blenders Text Editor.

In the Text Editor name the file **ran.py**. You will copy and paste this into a Blender file. You can type it directly into the Blender text editor but having it saved as a text file gives you a back up.

Note: Make sure the text is copied **exactly** as shown. The slightest error will cause an error when running the script and importantly include the **.py** suffix at the end of the File Name.

Register The Script

With the Text written in the Text Editor in the Blender File, the next step is to register the Script. Registering the Python Script makes functions in the script (**randf**) available for use by the Blender Drivers. Functions can be called (entered into Blender) for use by a Driver.

In the **Text Editor Header** click on **Text** and at the bottom of the menu that displays click **Run Script.** This is a check to make sure you haven't made an error. If the Script is OK nothing happens but if there is a mistake you receive an error message (Figure 30.7). Check everything and try again.

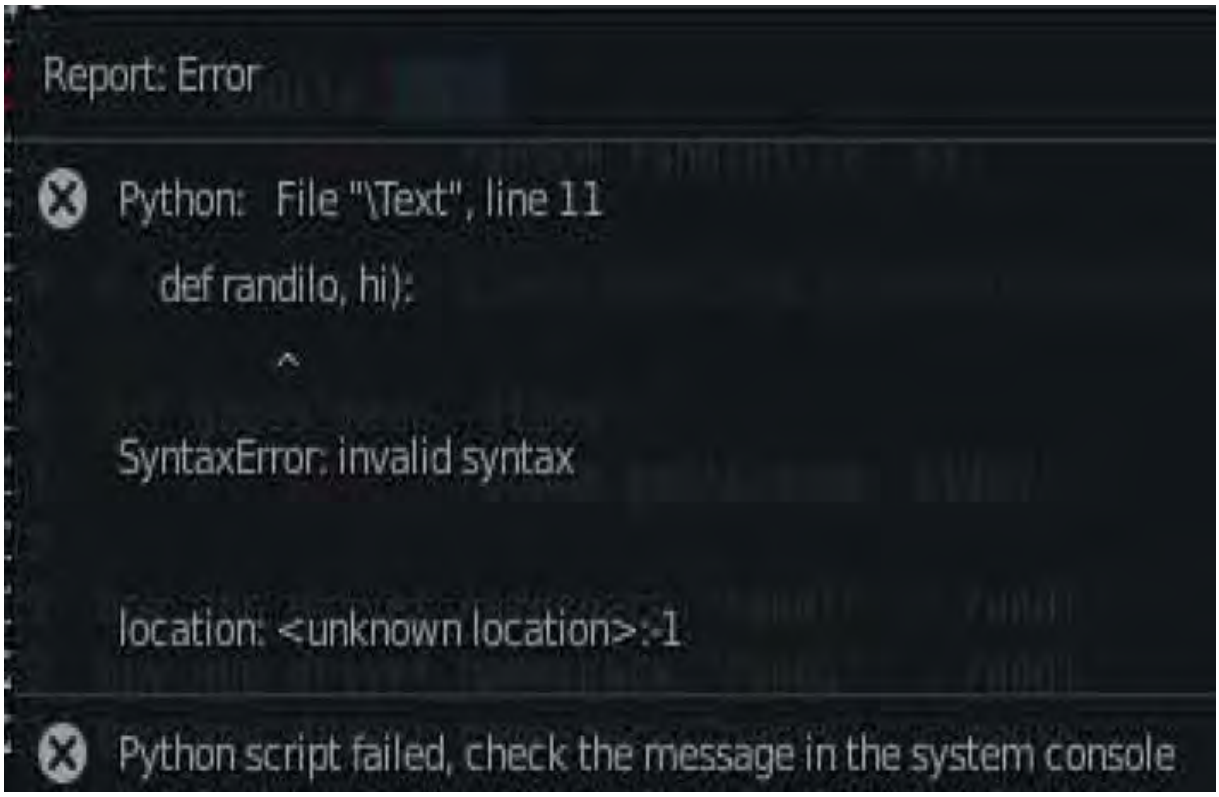


Figure 30.7

Having run the Script with no errors, click **Text** again in the Header and **check Register** to enter the Script in the Blender File.

Registering the Python Script makes functions in the script (**randf**) available for use by the Blender Drivers. Functions can be called (entered into Blender) for use by a Driver.

At this point save the Blender File. Name the File **RandomPy.blend**.

You now have a Blender File available for generating random properties. How the **randf (lo, hi)** part of the Python Script works will be demonstrated. To understand this statement you will have to undertake a study of the Python programming language.

Have the **RandomPy.blend** file opened. You can close the **Text Editor** and return the **3D Viewport**.

Set Up a Workspace

With the mouse cursor in the **3D Viewport Editor** press the **N key** to display the **Object Properties Panel** at the RHS. **Note:** This Properties Panel is distinct from the **Properties Editor** displayed at the RHS of the Screen.

Note: Some values display in both i.e. in the **Properties Editor, Object Properties, Transform Tab** there are **Location, Rotation and Scale** sliders. Identical sliders for the same Transform properties are displayed in the Object Properties panel of the 3D View Editor.

Split the **3D View Editor** in two and change one part to the **Drivers Editor** (Figure 30.8). The Driver Editor, by default, comprises two panels. An empty Driver panel at the LH side and a Graph panel on the right.

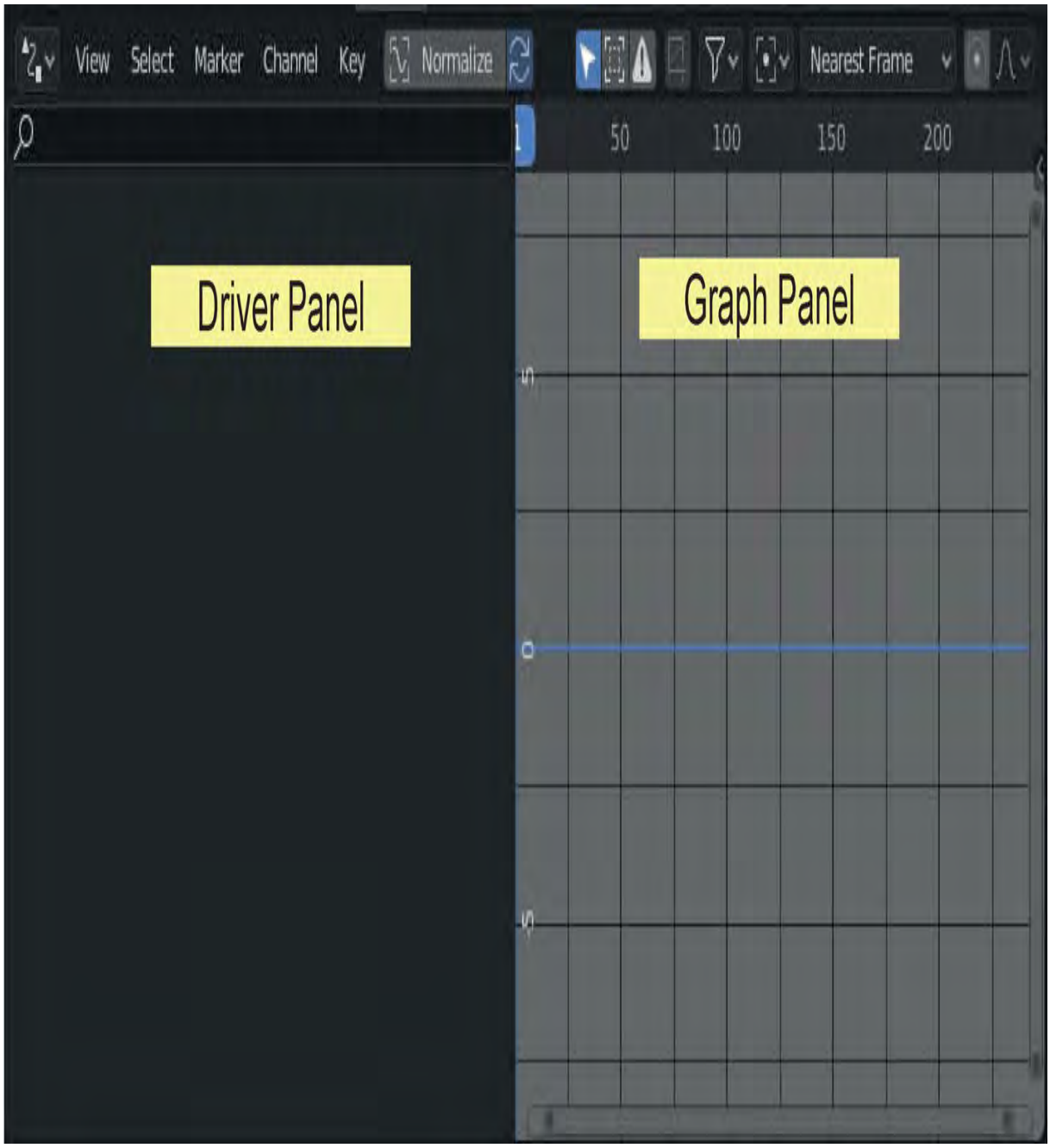


Figure 30.8

Note: As an alternative to setting up a Workspace you may select the **Scripting Workspace** in the Blender Screen Header. Creating a personalised Workspace has the advantage of familiarising you with the controls.

With the Mouse Cursor in **either panel** of the Driver Editor, press the **N Key** to display the **Properties Panel** with **Show Cursor** at the top.

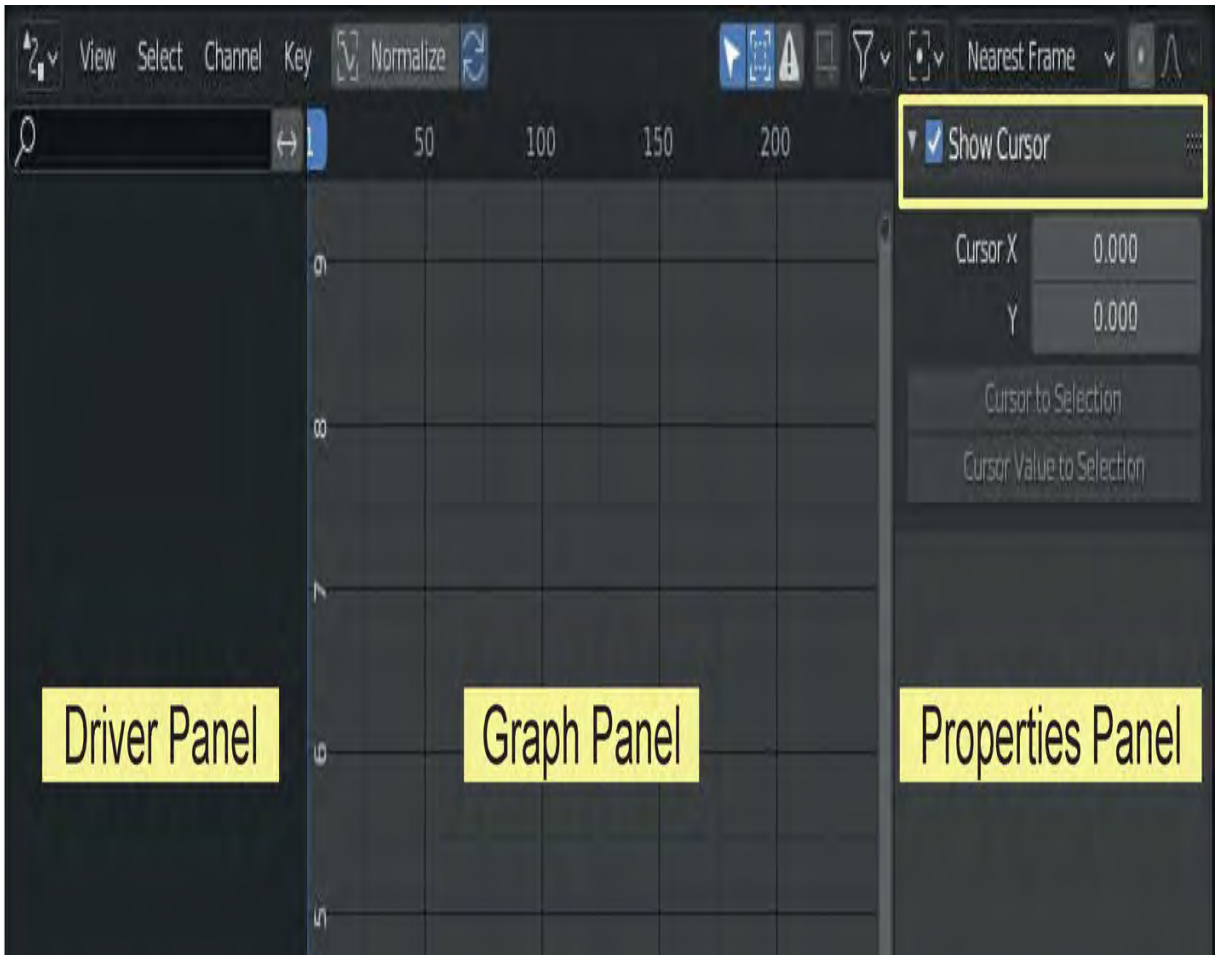


Figure 30.9

The Graph Panel will not be used in this exercise, so you can arrange the Editor to only show the **empty Driver panel** and the new **Properties Panel** (Figure 30.10).

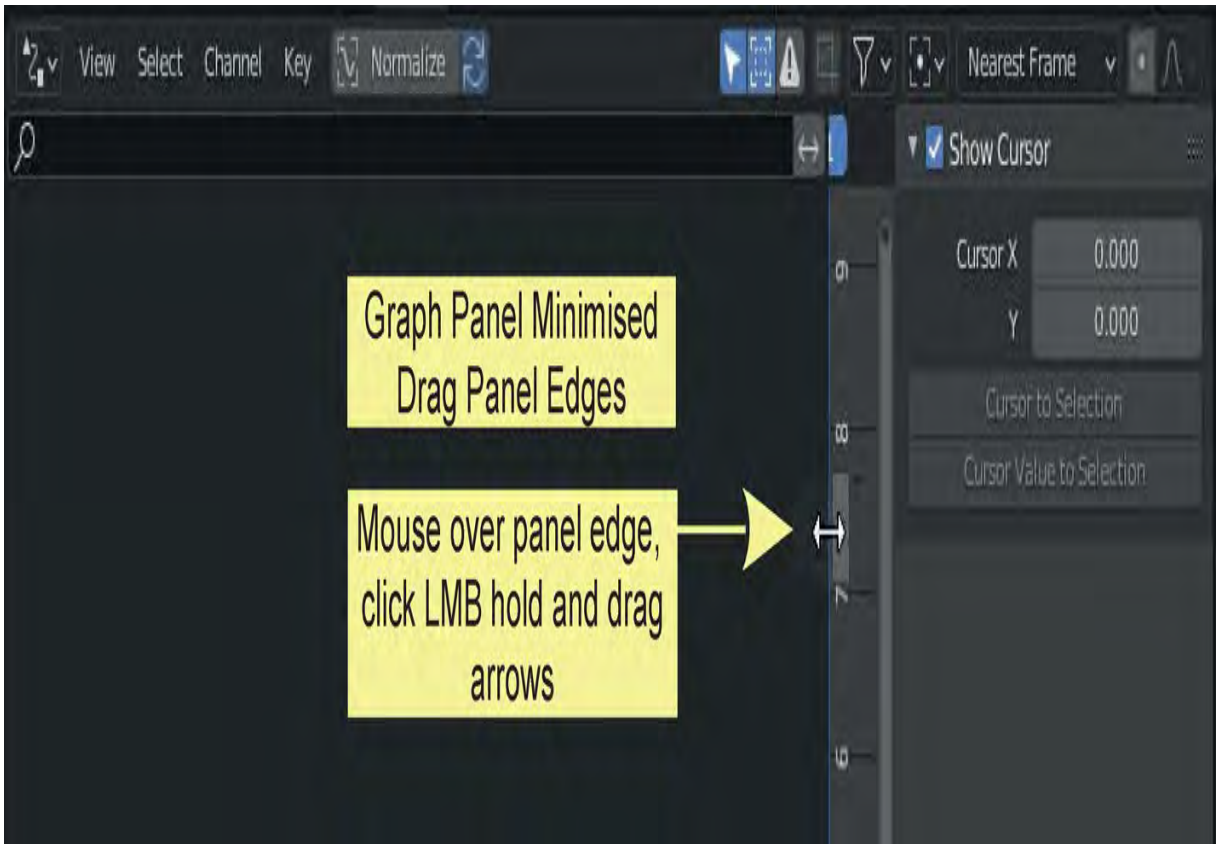


Figure 30.10

Drag edges of panels leaving the Graph display as a narrow strip.

Create the Blender File

With the Graph Editor configured save the Blender file. In creating the file you have a registered **Python Script** in the file and have configured the 3D View Editor. Remember, the file is named: **RandomPy.blend**. (You can name the file anything you like. Just remember what the name is and where you save it.).You may close the Blender file in which case you have a file set up for future use.

30.5 Using The RandomPy.blend File

It is assumed the **Blender File** named **RandomPy.blend** incorporating the **Python Script** named **ran.py** has been saved and Blender has been closed. Open Blender and open the **RandomPy.blend** Blender file.

Rotation Driver Properties

The will be used to make the default Cube Rotate in the 3DViewport Editor

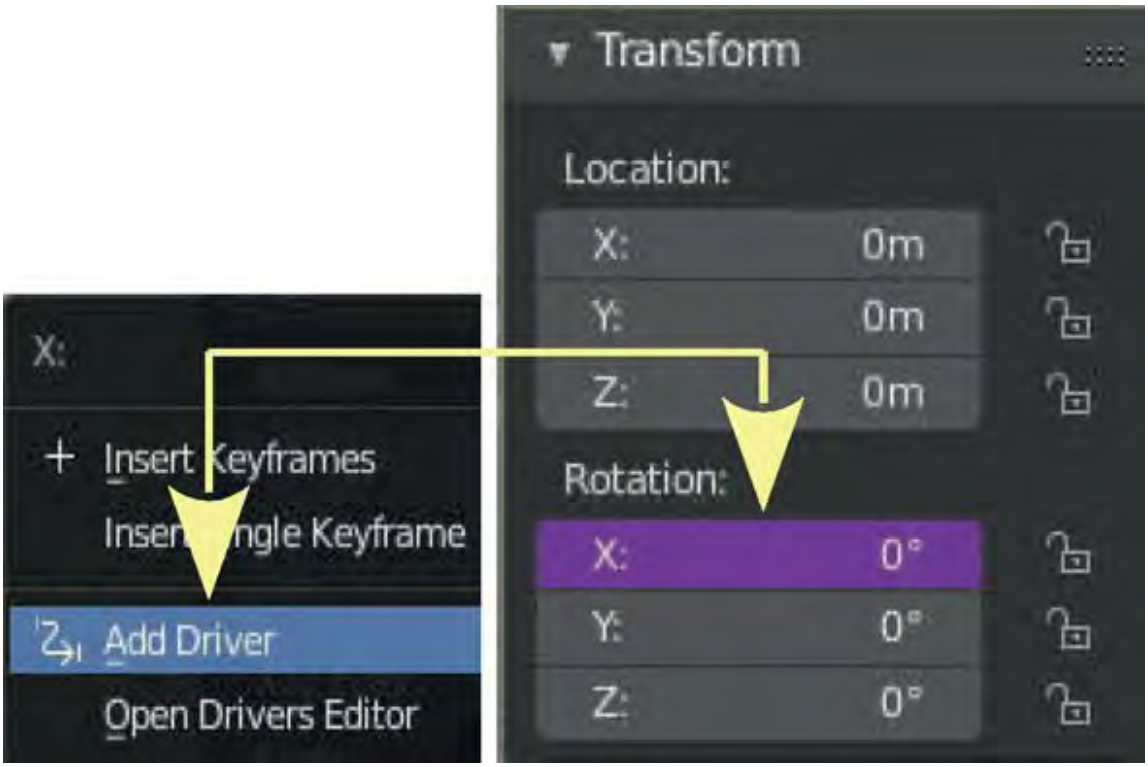
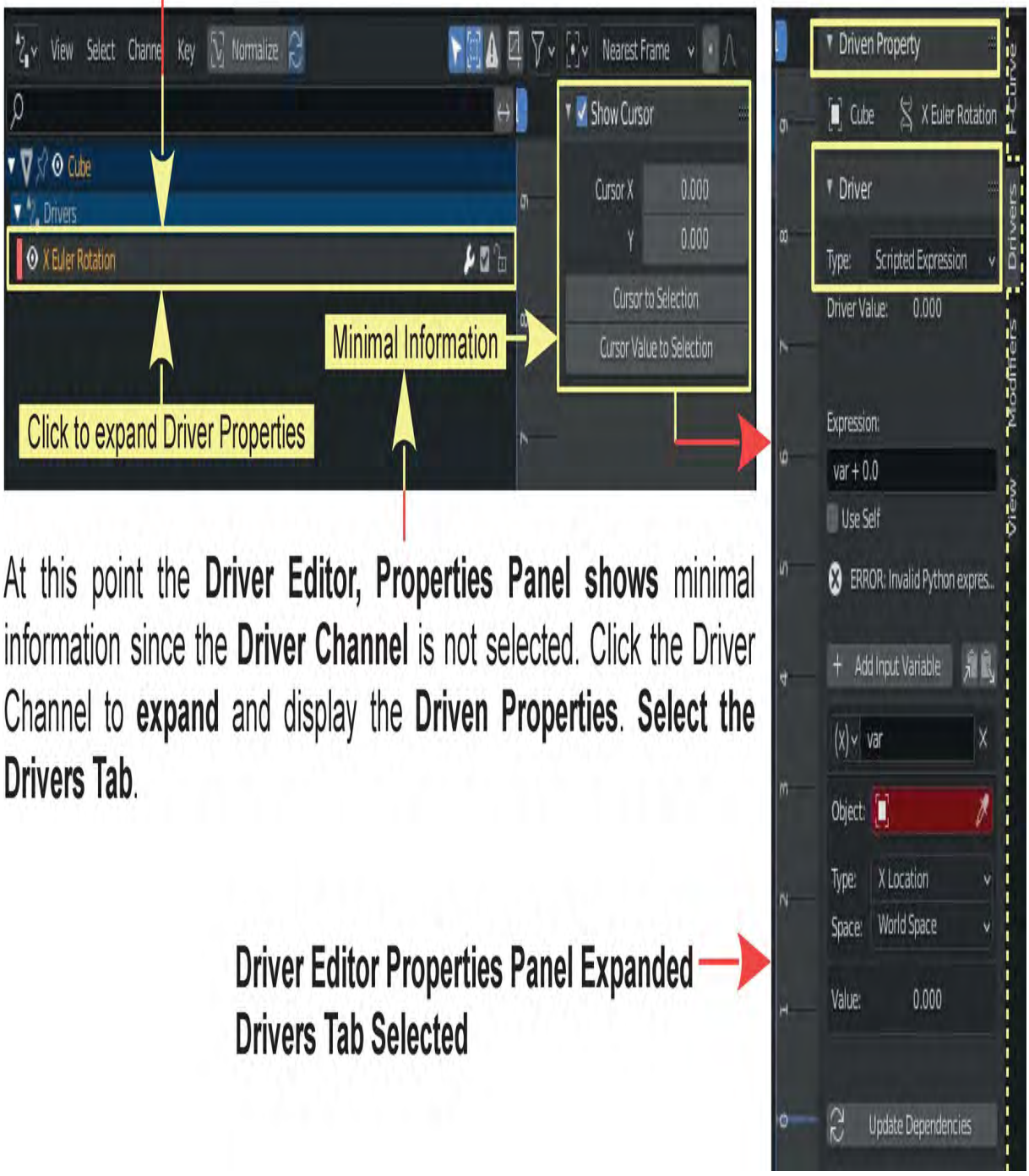


Figure 30.11

Begin by setting a Driver for the **X Rotation Property** of the default Cube Object. **Make sure the Cube is selected**. In the **Object Properties Transform Panel** of the **3D Viewport Editor**, right mouse click on the **XRotation slider** and select **Add Driver** (Figure 30.11).

The slider turns purple indicating a driver has been added and the **Driver Channel** displays in the **Driver Editor** (Figure 30.12).



At this point the **Driver Editor, Properties Panel shows** minimal information since the **Driver Channel** is not selected. Click the Driver Channel to **expand** and display the **Driven Properties**. **Select the Drivers Tab**.

Driver Editor Properties Panel Expanded
Drivers Tab Selected

Figure 30.12

At this point the **Driver Editor, Properties Panel shows** minimal information since the **Driver Channel** is not selected. Click the Driver Channel to **expand** and display the **Driven Properties**. **Select the Drivers Tab**.

In the Driver Tab you see (by default) the **Type: Scripted Expression**. This is telling the Driver to use the Python Script entered in this Blender File.

Below the **Driver Type** you see **Expression** with **var + 0.0** entered and below this you see the **ERROR: Invalid Python expression** since **var + 0.0** are incorrect values for the Scripted Expression (the Python Script). The values are called Argument in computer language which tells the computer to use the values as the criteria for performing calculations in the Python Script (the Scripted Expression). Since you are concerned with the X Rotation of the Cube appropriate values are -pi and pi. These values refer to the Greek **rr** which is approximately equal to 3.142, therefore, by entering – pi and + pi you are saying use the values Plus minus and plus 3.142.

Enter the Python Script Function

In the **Driven Property, Expression** enter **randf(-pi, pi).**

Run the Script

You run or execute the Script by clicking the **Update Dependencies** button at the bottom of the Driven Property Panel. At each click the Script randomly recalculates the rotation of the Cube in the 3D Viewport Editor.

More Drivers

You may add drivers to the Cube i.e. set up a **X Axis Location Driver** by the same procedure but entering numeric values in the **Driven Property Expression i.e. randf(0, 10).**

In this case the X Axis Location of the Cube is also randomly recalculated at the same time as the X Axis Rotation.

Drivers may be applied to other Properties such as Scale and even Materials (Color).

Re-evaluating Drivers

With drivers set, in the 3D window solid shading display mode, press G key (Grab) drag the mouse and move the object in the 3D window. Blender constantly re-evaluates the driver and produces random values for the properties values. Note this will effect any driver which has the “randf”, “randi”, or “gauss” python expression inserted. This is just another method of updating dependencies. This function can be negated by clicking on the camera icon in the driver dope sheet panel adjacent to the driver (Mute). You will want to negate this function if you require to manually reposition the cube.

OK! Take a deep breath. You are not finished.

Note: If you take a break and save and close the program, then come back to it later on, you may find this error message: Click **Allow Execution**.

Note: You may also find **Invalid Python Script ERROR** message in the Driven Property Panel in the Driver Editor. Allowing Execution of the Script cancels, if not open the Text Editor window which will have the Python script and click Run Script.

30.6 Material Property Drivers

Material properties can be randomized using Python Script just as you did with the rotation and Translation properties. For the rotation you used a value range of **-pi** to **pi** (2rr radians in a circle, therefore -rr to +rr) and for the Translation the range was 0.0 to 10.0 Blender units).

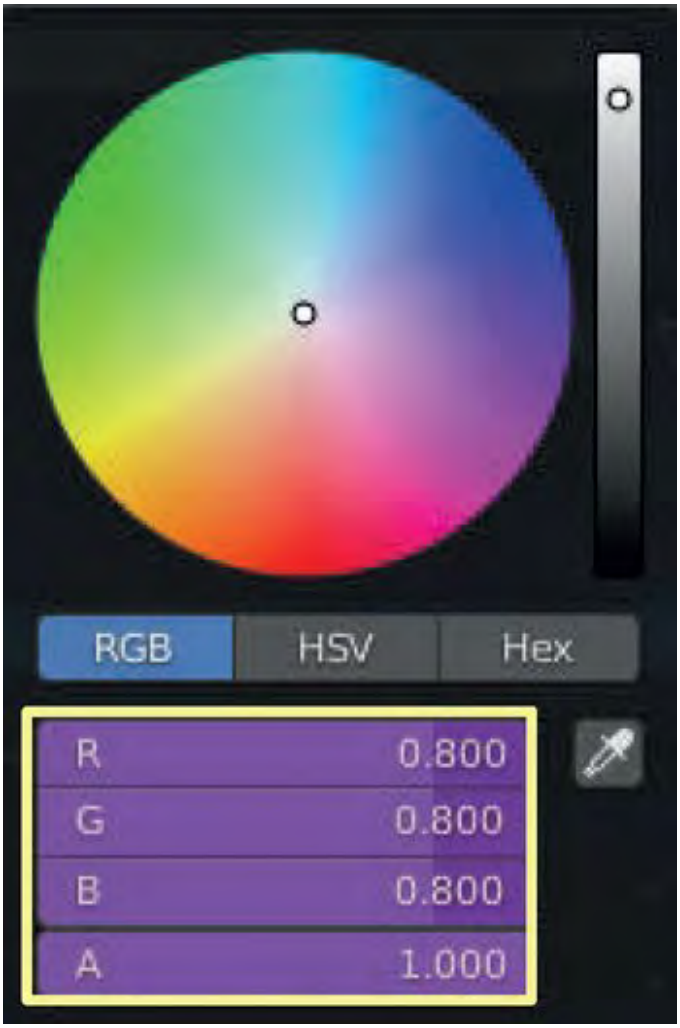


Figure 30.13

Consider the Expression **randf(min value, max value)**. What do you use here for Material color? Take a look at the Diffuse material color sliders in the color picker in the **Properties Editor Material Properties** . There are three sliders for the red, green and blue color channels each with a minimum value of 0.000 (black) and a maximum value of 1.000 (white) (Figure 30.13). The default value is 0.800 (gray),

The values between these maximum and minimums produce the spectrum of visible color between white and black. To randomize material within the spectrum all that is required is to use the expression **randf(0.000, 1.000)**.

Figure 30.14

Right click on the **Base Color** bar and select **Add Driver**. A purple outline displays around the color bar in the Properties Editor (Figure 30.14) and a Material Drivers Channel is displayed in the Driver Editor. (Figure 30.15).

View
Select
Channel
Key
Normalize
Cube
Drivers
X Euler Rotation
Material
Shader Nodetree
R Default Value (Base Color)
G Default Value (Base Color)
B Default Value (Base Color)
A Default Value (Base Color)
Driven Property
Shade...
B Default Val...
Driver
Type:
Scripted Expression
Driver Value:
0.800
Expression:
var + 0.8
Use Self
ERROR: Invalid Python expres...
Add Input Variable
var
Object:
Type:
X Location
Space:
World Space
Value:
0.000
Update Dependencies
F-Curve
Drivers
Modifiers
View

Figure 30.15

With the B – Color Channel Selected (highlighted yellow) the Driven Property Panel displays the Driver Properties for the B Component of the Base Color.

To randomize the color, replace **var + 0.8** in the **Expression** with:

Randf(0.000, 1.000)

With each click on **Update Dependencies** the color of the Cube changes in the 3D Viewport Editor.

Note: Have the 3D Viewport Editor in Material Preview or Rendered Viewport Shading Mode.

Repeat for the R and G Color Channels.

30.7 Duplicating the Object

In the 3D window you only have a single Cube in the Scene. To create a bunch of cubes scattered about the Scene you could select the Cube in the 3D Viewport Editor and simply press Shift + D Key to duplicate the Cube.

A duplicate is created named Cube.001 and placed in grab mode ready to be translated. Drag the mouse to relocate and observe that the original and the duplicate are both re-evaluated by the drivers and the properties change.

There are now two Cube objects in the scene which are identical and appear to be separate. The duplicate Cube (Cube.001) is however linked to the original (Cube) in that it is sharing properties data-blocks of the original. If you translate **Cube.001, Update Dependencies** is activated and the properties of the cube change. If you translate **Cube** the properties of both cubes are changed. The Material color of both Cubes is the same.

Shift select both cubes and you will see two sets of drivers in the **Drivers Editor** (Figure 30.16).

View
Select
Channel
Key
Normalize
Cube
Drivers
X Euler Rotation
Material
Shader Nodetree
R Default Value (Base Color)
G Default Value (Base Color)
B Default Value (Base Color)
A Default Value (Base Color)
Cube.001
Drivers
X Euler Rotation
Material
Shader Nodetree
R Default Value (Base Color)
G Default Value (Base Color)
B Default Value (Base Color)
A Default Value (Base Color)

Figure 30.16

To make the Cubes display color independent of each other, shift select both, click Cubes in the 3D Viewport Editor Header. Select **Relations – Make Single User – Materials** (Figure 30.17).

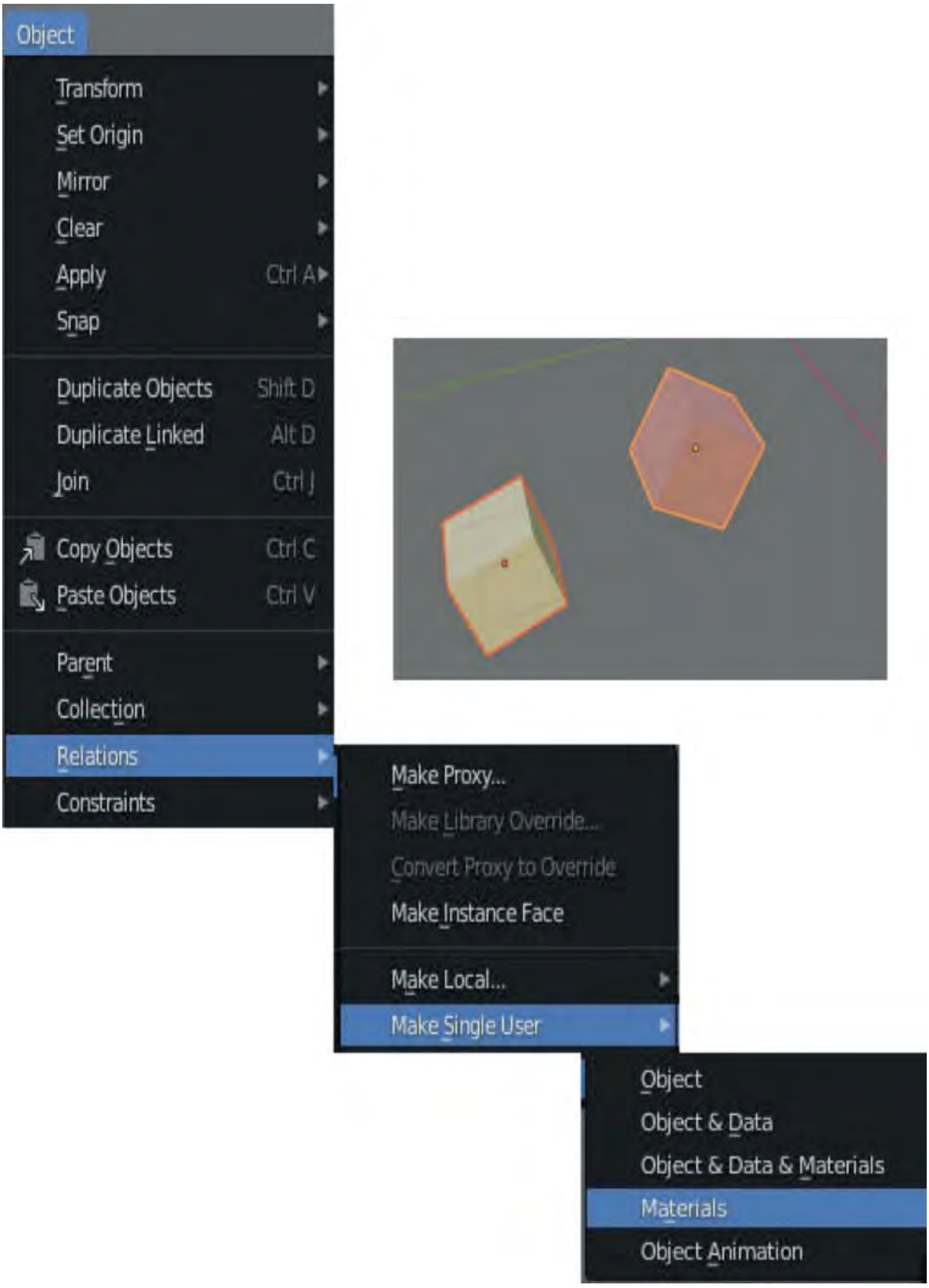

Object
Transform
Set Origin
Mirror
Clear
Apply Ctrl A
Snap
Duplicate Objects Shift D
Duplicate Linked Alt D
Join Ctrl J
Copy Objects Ctrl C
Paste Objects Ctrl V
Parent
Collection
Relations
Constraints
Make Proxy...
Make Library Override...
Convert Proxy to Override
Make Instance Face
Make Local...
Make Single User
Object
Object & Data
Object & Data & Materials
Materials
Object Animation

Figure 30.17

Select Cubes and repeat the duplication process as many times as you like (Figure 30.18).

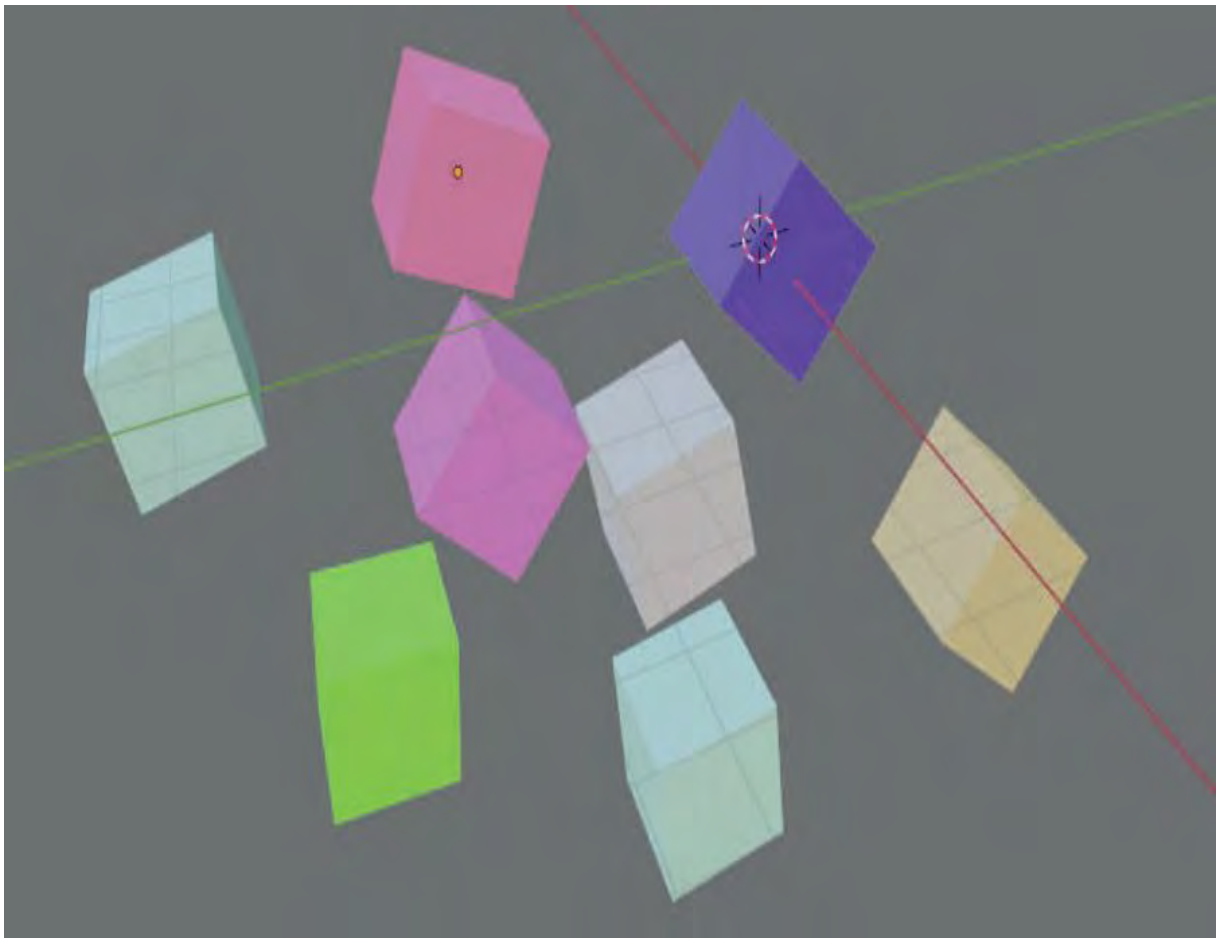


Figure 30.18

31 Cycles & Workbench Render

31.1 Cycles Render
31.2 How to Start Cycles
31.3 Create an Object Light Source
31.4 Cycles in Practise
31.5 Workbench Render

The Render Engines

There are three **Render Engines** built into Blender, **Eevee, Cycles** and **Workbench**.

Eevee is the default Engine which provides several Viewport display options one of which is a Real Time display showing what is produced in a final Render.

The **Cycles** Rendering system, built into Blender, is designed to produce photo realistic images and to provide an interactive workspace where you see a rendered view as a Scene is created.

Photo realism and high definition in images, including animation frames comes with a demand on computer power and render time.

The Eevee Render Engine gives an excellent result but there are situations where Cycles will provide added benefit.

All the Blender tools and controls for generating a Scene are applicable to both Render processes although there is a difference in the Node Systems.

Workbench Render provides a simplified process for previewing Scenes in the construction process before implementing a final Render.

Creating Scenes and Rendering,will be described in this chapter limiting the instruction to how to start the Engines, obtain the best results from your computer and provide a brief example of applications.

31.1 Cycles Render

Cycles Rendering simulates many effects that have to be specifically added to other methods of rendering such as soft shadows, depth of field, motion blur, caustics, ambient occlusion and indirect lighting.

The **Cycles Render Engine** is described as being a raytracing based engine with support for interactive rendering. Being interactive means you see a rendered view of your work as it progresses in the 3D Viewport Editor in **Rendered Viewport Shading Mode**. Cycles incorporates a Shading Node system, a different material and texture work flow and it utilises **GPU** acceleration.

Computer Specifications for Cycles

Before using Cycles, be aware that you will require a reasonable computer processor and a graphics card which meets the specifications to handle this advanced process (refer to the Blender Wiki – Hardware Requirements).

In essence, to fully utilise **Cycles** you need a fast processor, heaps of memory (RAM) and a graphics card with **Open GL** (graphics card with built in memory **GPU** and **CUDA** enabled).

> **Note:** Cycles Rendering is activated from the main Blender interface but CUDA and GPU acceleration require a secondary activation similar to an Add-on.

If you are new to Blender these terminologies and specifications may be slightly on the technical side but just be aware that, to utilise the full effects of

Cycles you computer has to meet the requirements. The following will show you how to activate Cycles and discover if your system is up to speed.

31.2 How To Start Cycles

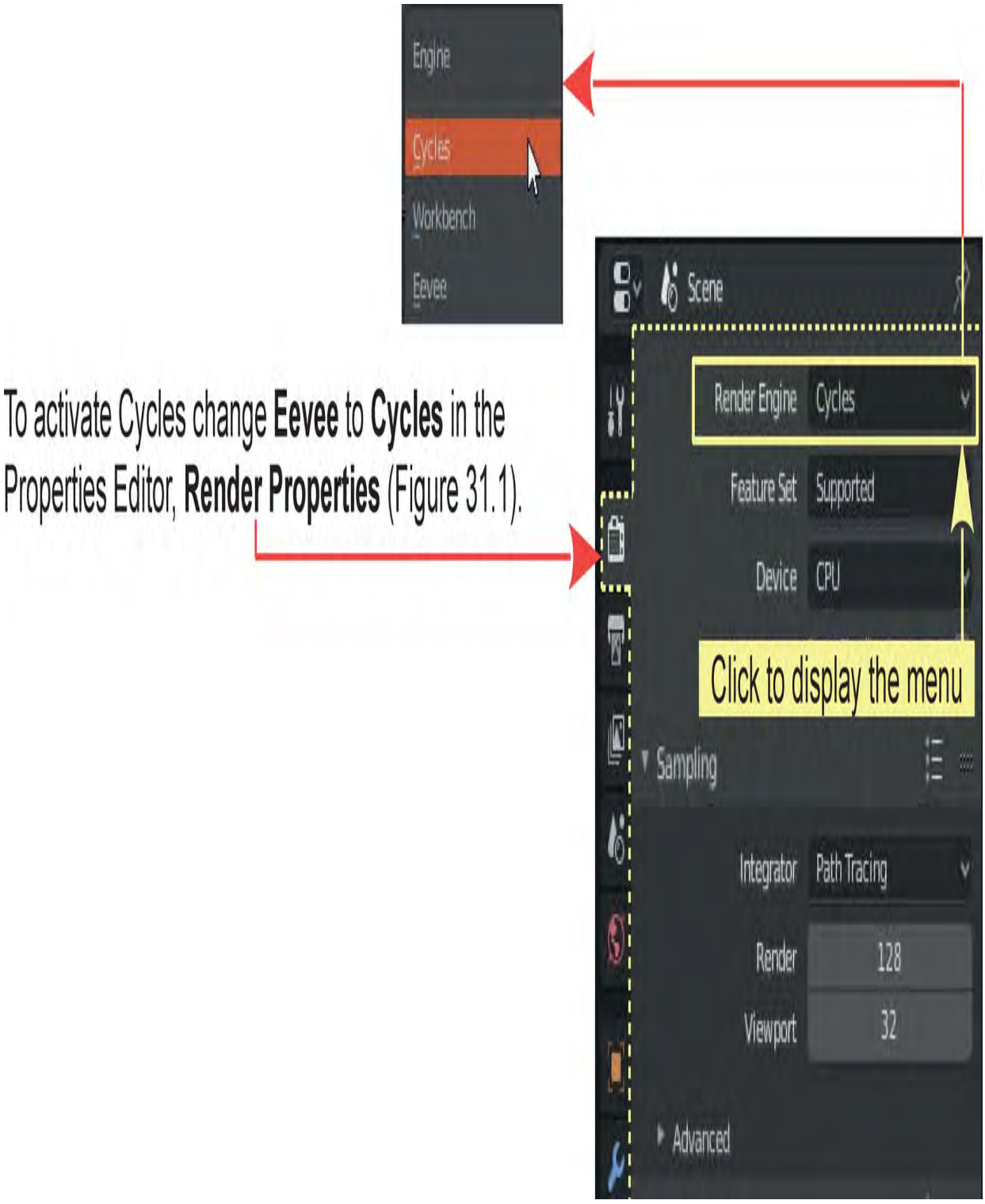

Engine
Cycles
Workbench
Eevee
To activate Cycles change **Eevee** to **Cycles** in the Properties Editor, **Render Properties** (Figure 31.1).
Scene
Render Engine
Cycles
Feature Set
Supported
Device
CPU
Click to display the menu
Sampling
Integrator
Path Tracing
Render
128
Viewport
32
Advanced

Figure 31.1

To demonstrate Cycles Rendering set up a Scene in the 3D Viewport Editor as shown in Figure 31.2.

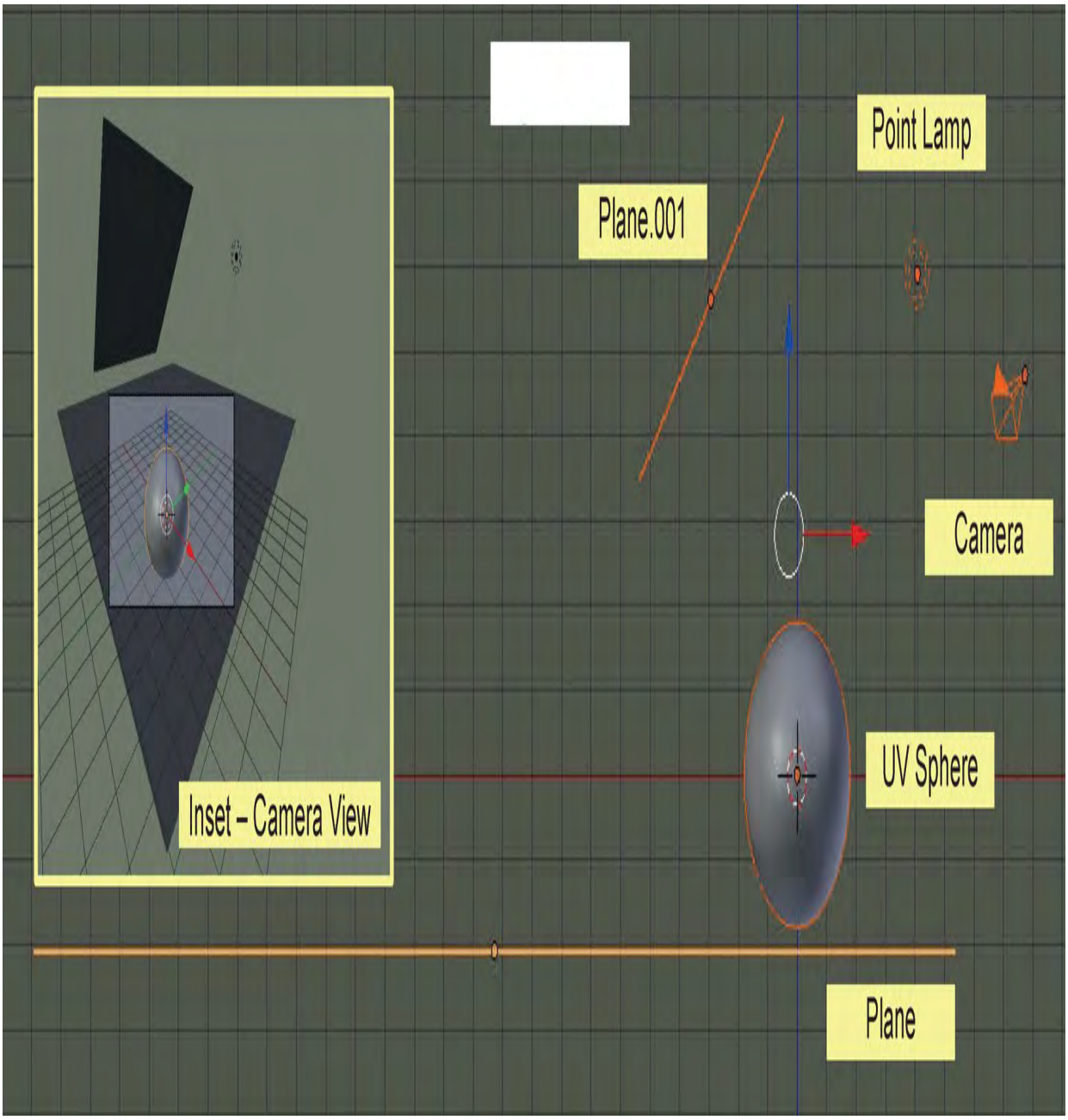


Figure 31.2

Figure 31.2 shows a UV Sphere positioned just above a Plane (ground plane). A second Plane (Plane.001) is placed above and behind the UV

Sphere opposite the Camera. The ground plane is scaled to fill the Camera aperture as shown in the Camera View inset, while Plane.001 is positioned outside Camera View. The UV Sphere is scaled up and set to Smooth. There is a single Point Lamp in the Scene.

To activate the **Cycles Render Engine** click on the selection button in the Properties Editor, Render Properties as shown in Figure 31.1. From the menu select **Cycles Render**. With the 3D Viewport Editor in **Rendered Viewport Shading Mode** you see a rendered view of the Scene.

You will note that some of the options in the **Properties Editor** change. Leave all the settings just as they are for this demonstration.

The 3D Viewport Editor will change showing the Objects with a dark gray background (Figure 31.3). If you rotate the Scene you will see the Objects being re-rendered as you rotate.

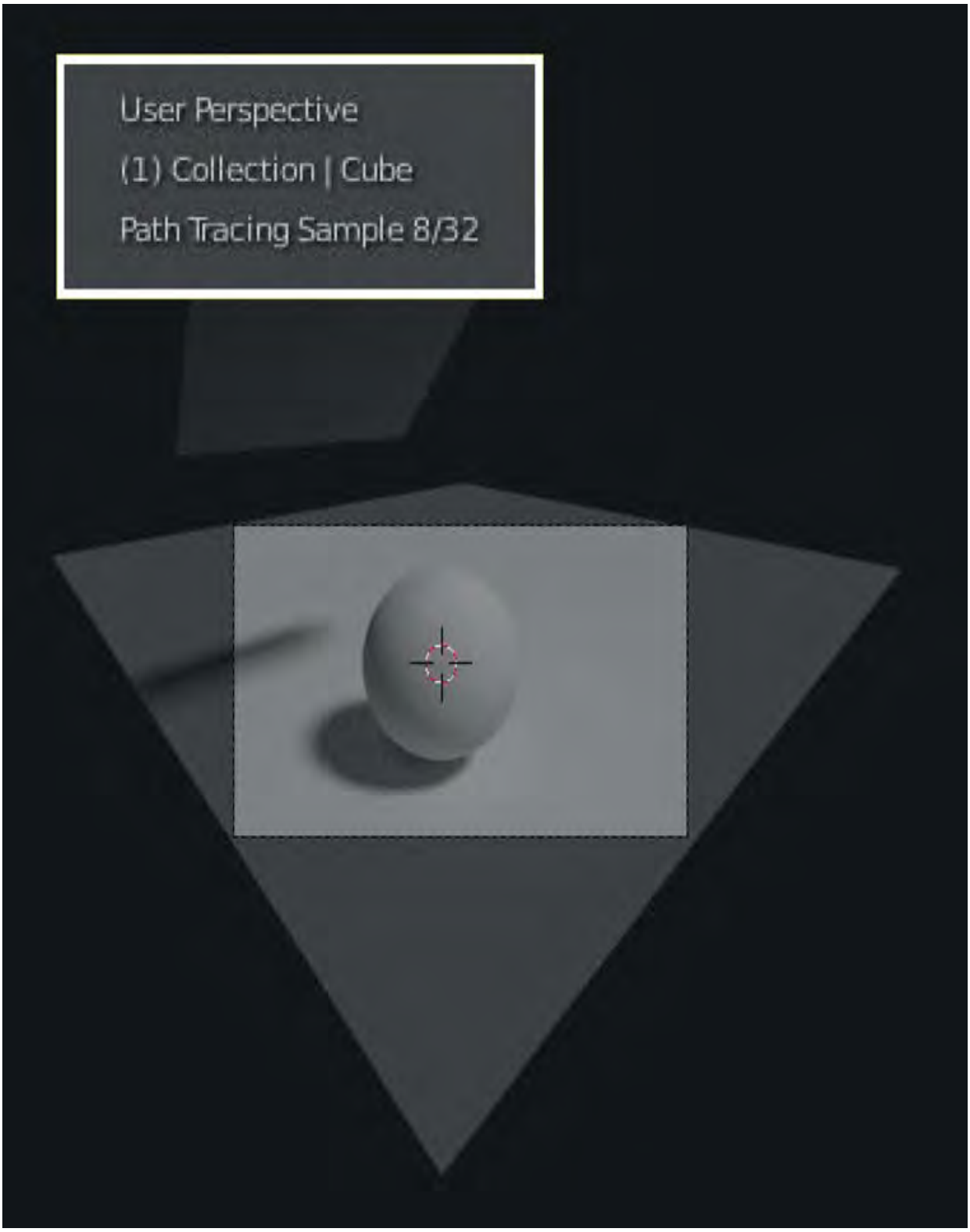

User Perspective
(1) Collection | Cube
Path Tracing Sample 8/32

Figure 31.3

Unless you have a reasonably fast processor the render will be very blocky and grainy and can take a considerable time. With Rendered Viewport Shading the 3D Viewport Editor re-renders at each change made to the Scene. The longer you wait the clearer the render becomes, **up to a point**. You wouldn't want your computer stuck in an infinite rendering loop so Blender incorporates a time-out setting to limit therender. When the Scene is altered e.g. rotated, the **Render** process is activated. In the upper LH corner of the 3D Viewport Editor you will see a progress display giving the number of Render Samples. When a Render is completed this displays, **Rendering Done**.

In the **Properties Editor, Render Properties, Sampling Tab** see **Render 128 – Viewport 32**. These are the default Render Pass settings. The render it limited to 32 Passes for the view in the viewport and 128 Passes for a Rendered View when you press F12.

Increasing the Pass settings produces a better result, bearing in mind that an increase in Samples incurs an increase in time to perform the Render.

The content of the Scene also affects the Render time. For example, on a machine, with the **Default Blender Scene** containing the single **Cube** Object the elapsed time for the 32 Sample preview is 00.02.20 seconds. The full image Render, pressing F12 with 128 samples takes 00.09.51 seconds. Scaling the **Cube** X 2 increases the preview time to 00.03.43 seconds and a full render is 00.24.89 seconds. **Note:** These Render Times are on my PC. Yours will be different.

The 3D Viewport Editor view may be very dark since the default Lamp provides only a limited illumination. Changing the Lamp type or providing additional Lamps will improve the illumination, but in Cycles you can use Objects as a light source. More on that later.

What to expect from Cycles will depend on your computer and operating system, your display adaptor (Graphics Card) and the drivers (Software) that have been installed for the card.

Before proceeding it will help to understand some terms.

NVIDIA graphics: NVIDIA is one of many suppliers of graphics chipsets used in graphics cards. At the time of writing Blender is configured to use NVIDIA with Open CL and CUDA or OptixX enabled for GPU rendering.

Open CL is a set of graphics standards used world wide which is designed to give maximum performance on the GPU.

GPU (Graphics Processing Unit) is the processing device built into the graphics card which performs computations in parallel with the computer's central processing unit (CPU).

CUDA™ (Computer Unified Devise Architecture) is a parallel computing platform and programming model that enables dramatic increases in computing performance by harnessing the power of the graphics processing unit (GPU).

OptiX is **a domain-specific API designed for accelerating ray tracing**. It provides a complete package with programmable ray generation, intersection and shading while using RT Cores on NVIDIA RTX GPUs for accelerating Bounding Volume Hierarchy (BVH) traversal and ray/triangle intersection testing.

To summarize, the GPU performs computations in conjunction with the CPU which significantly speeds up the changing graphics display that is required for **On the Fly** graphics rendering. BUT! whether the GPU is faster than the CPU depends on your computer configuration. It could be your CPU is faster for some aspects of the process.

If you have a NVIDIA GPU you can use either CUDA or OptiX. OptiX is specifically built for ray tracing and is probably faster than CUDA, which is built for general compute. Blender OptiX option uses NVIDIA RT tech, which is only available on the newer NVIDIA cards (NVIDIA GeForce, Quadro and Tesla products with Maxwell and newer generation GPUs.)

Another factor in this technicality is the **Compute Capability** rating of your graphics card. Cards are rated through a range something like 1.1 to 3.5. At the time of writing, Blender only supports graphics cards rated at 1.3 and above for GPU processing (Rendering), so again, unless your system meets the requirements you will not realize the full capability of **Cycles**.

OK! Cycles has been turned on and there is a display on the Screen but at this point the ancillary architectures are not activated. CUDA, OptiX and OpenCL require a secondary activation.

Also be aware that what you see in the Blender controls will depend on your system configuration. Blender takes a look at your system and displays controls accordingly.

To activate CUDA, OptiX or OpenCL, in the **Blender Screen Header** click on **Edit** and open the **Preferences Editor**. Select **System** in the LH column to see **Cycles Render Device** (Figure 31.4).

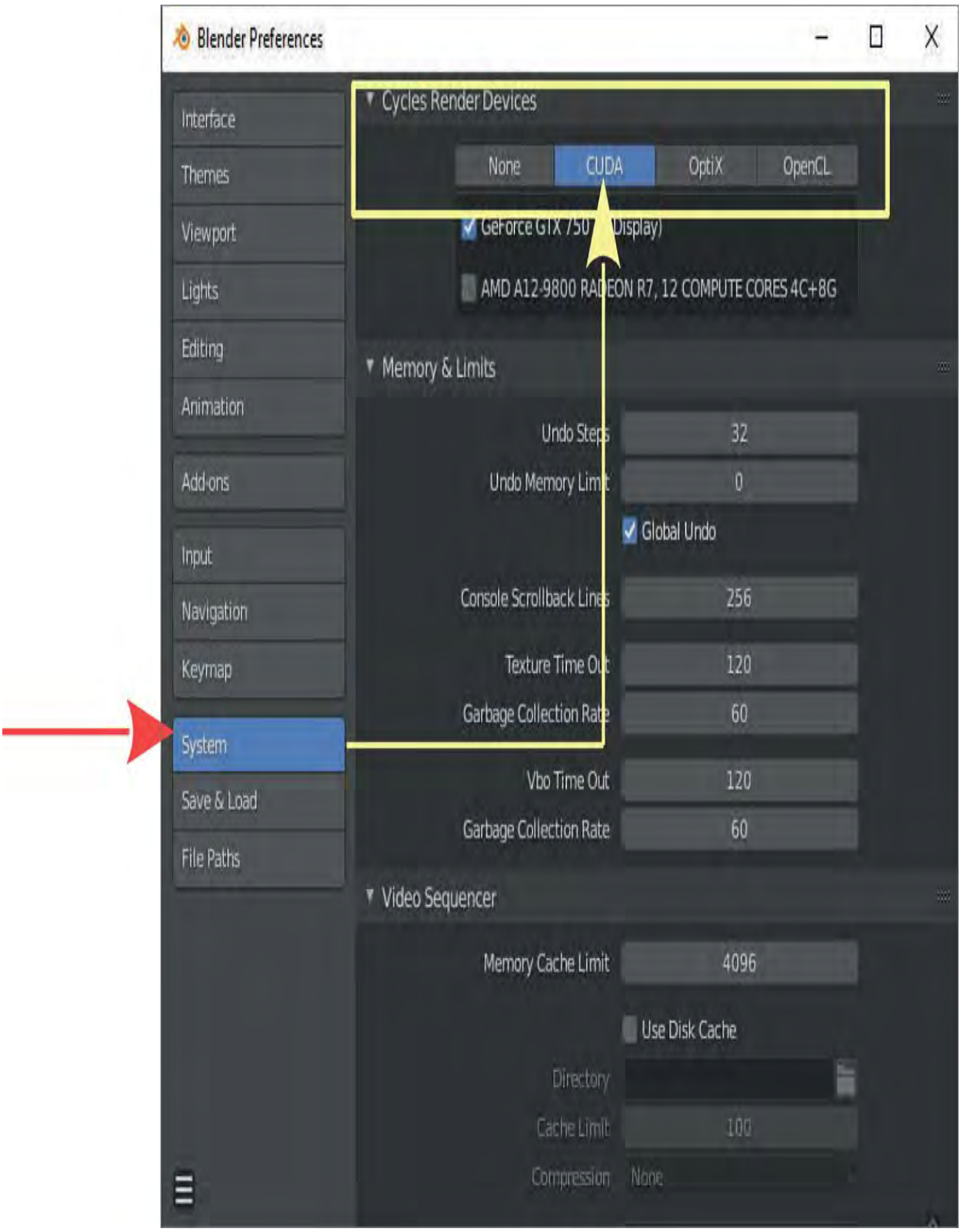

Blender Preferences
Interface
Themes
Viewport
Lights
Editing
Animation
Add-ons
Input
Navigation
Keymap
System
Save & Load
File Paths
Cycles Render Devices
None
CUDA
OptiX
OpenCL
GeForce GTX 750
Display)
AMD A12-9800 RADEON R7, 12 COMPUTE CORES 4C+8G
Memory & Limits
Undo Steps
32
Undo Memory Limit
0
Global Undo
Console Scrollback Lines
256
Texture Time Out
120
Garbage Collection Rate
60
Vbo Time Out
120
Garbage Collection Rate
60
Video Sequencer
Memory Cache Limit
4096
Use Disk Cache
Directory
Cache Limit
100
Compression
None

Figure 31.4

If you do not have a NVIDIA graphics chipset or your drivers for the card are outdated then you will see a message stating **No compatible GPUs** found and, therefore, the Cycles rendering process will be performed entirely by the CPU.

Providing you have the correct graphics chipset you may click on CUDA and the Cycles Render Devices Tab will show the name of your graphics card (Figure 31.5).

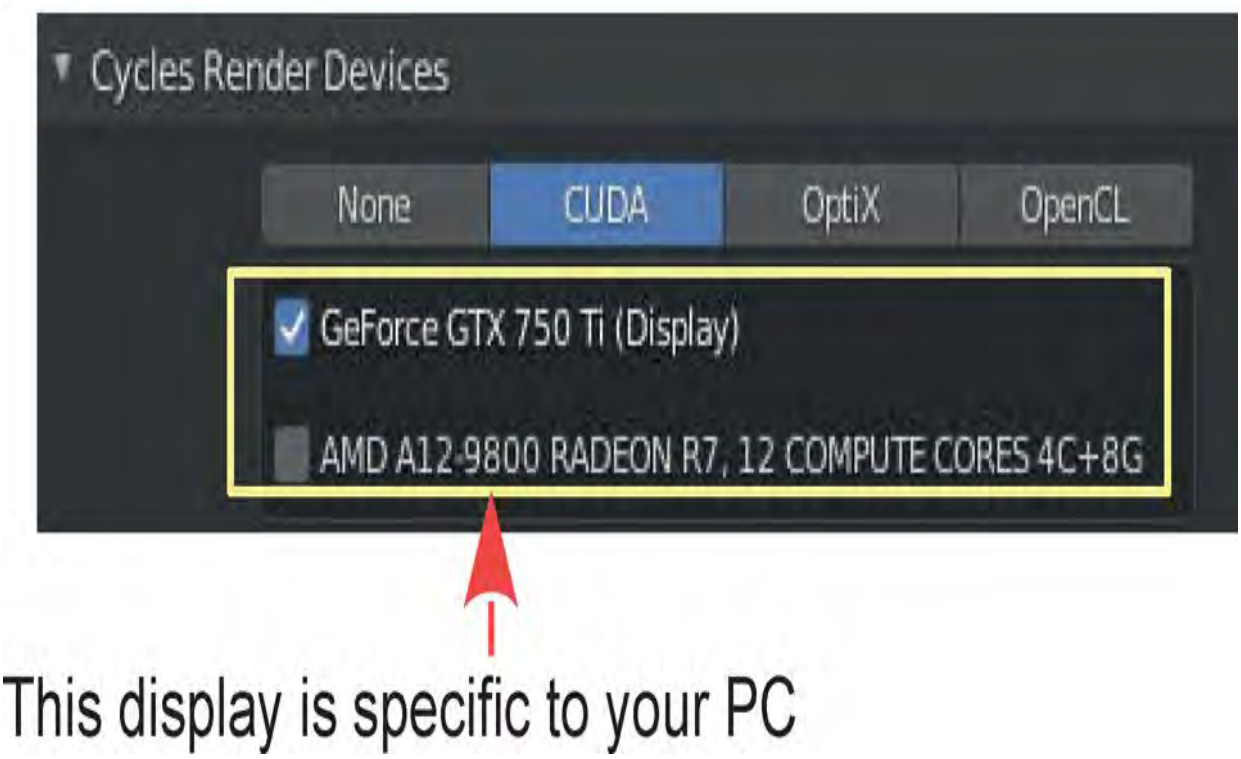


Figure 31.5

In the **Properties Editor, Render Properties** you will have the option to select **GPU Rendering** (Figure 31.6).

With Cycles activated it's time to see what it can do.

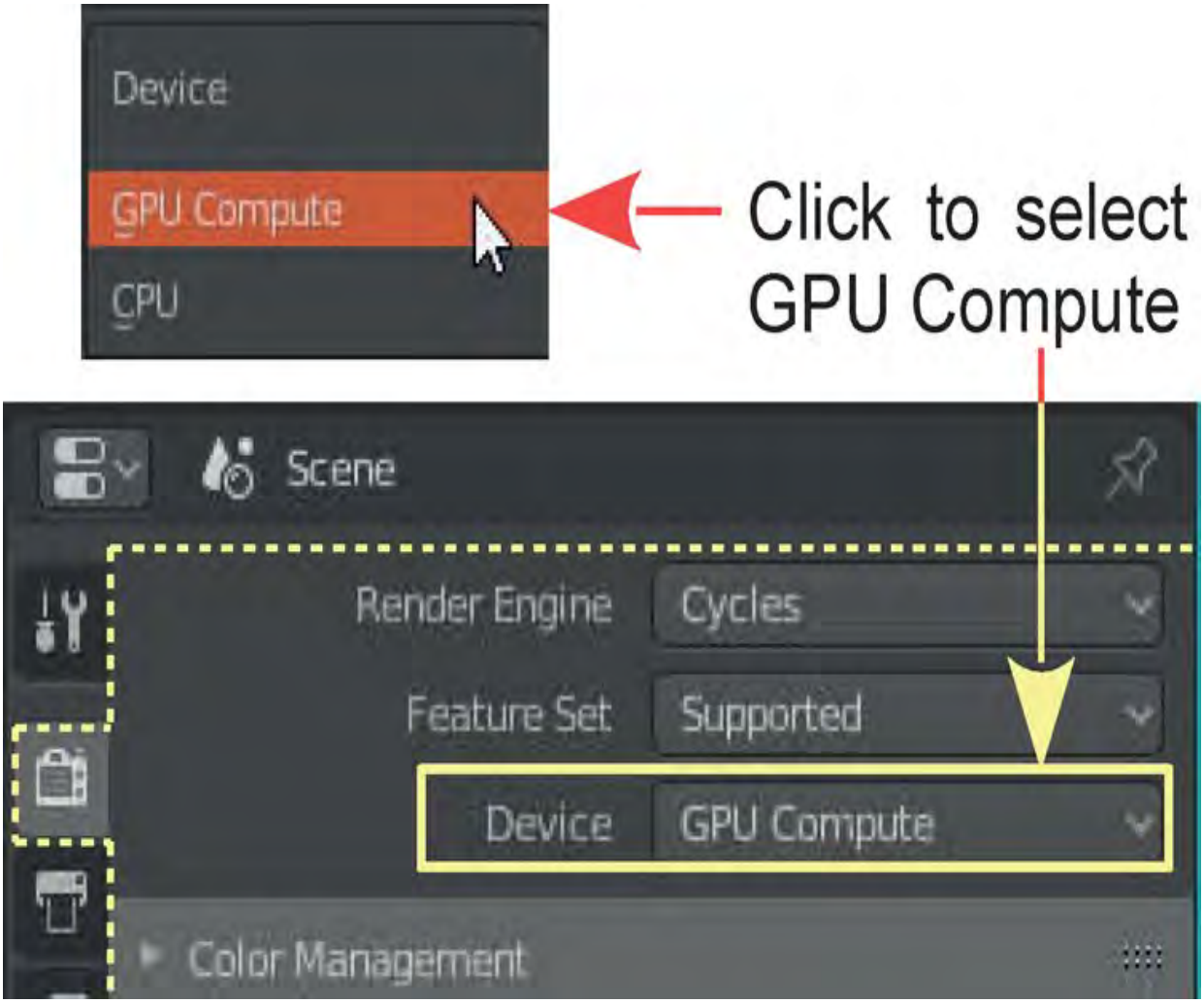


Figure 31.6

31.3 Create an Object Light Source

With the demonstration Scene created select the elevated Plane (Plane.001).

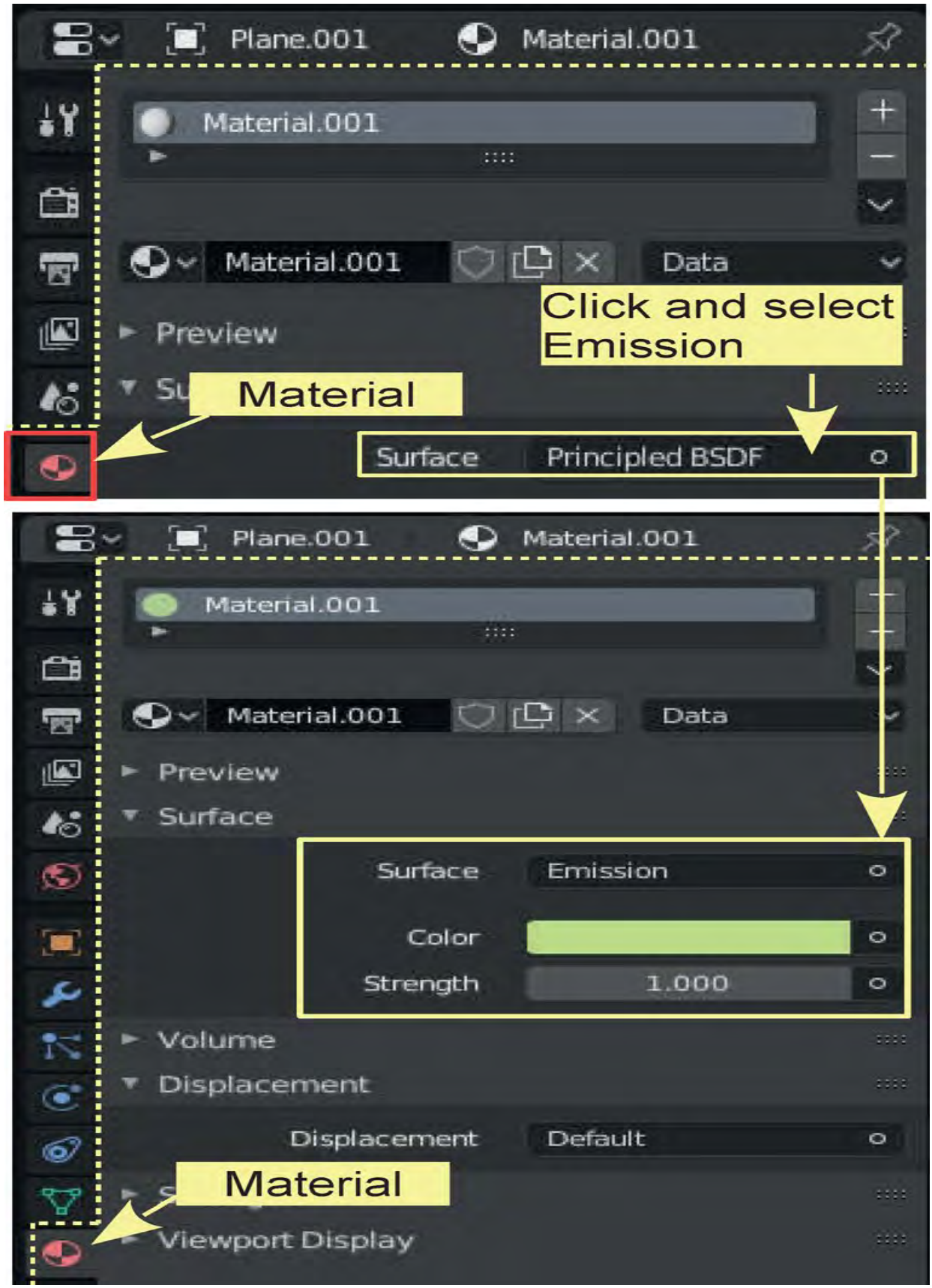
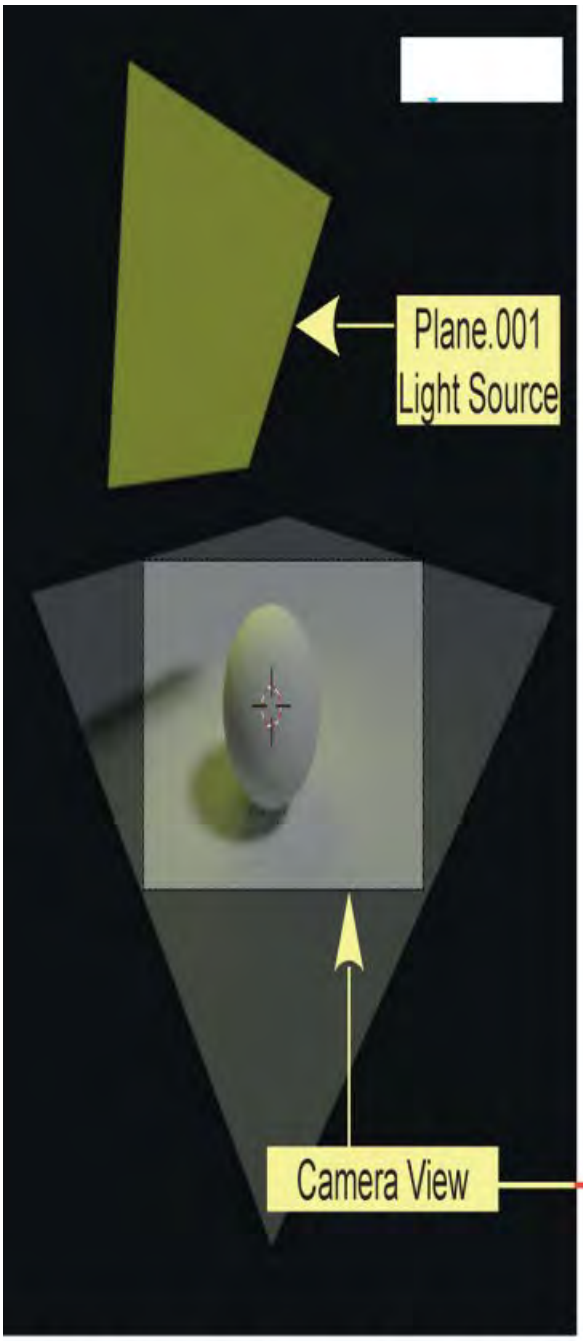

Plane.001
Material.001
Material.001
Material.001
Data
Preview
Click and select Emission
Material
Surface
Principled BSDF
Plane.001
Material.001
Material.001
Material.001
Data
Preview
Surface
Surface
Emission
Color
Strength
1.000
Volume
Displacement
Displacement
Default
Material
Viewport Display

Figure 31.7

Go to **Properties Editor, Material Properties**. **Note:** When the **New button** has been pressed the Materials buttons will show the **Base Color** (color) being controlled by the **Principled BSDF Node**. You may verify this by opening the Shader Editor where you will see the Principled BSDF Node connected to the Material Output Node.

In **Surface bar** where you see **Diffuse BSDF** click and select **Emission** from the menu that displays. This changes the plane into a light source (Figure 31.9). Note the value is 1.000 to increase thelight intensity).



Plane.001 Light Source – Green Light

Rendered Image

Figure 31.9

In the **Surface Tab** click on the **Color bar** and in the color picker that displays, select a bright color (green).

Plane.001 Light Source – Green Light

Plane.001 emits light casting it in the Scene where it reflects off the UV Sphere and ground Plane (Figure 31.10).

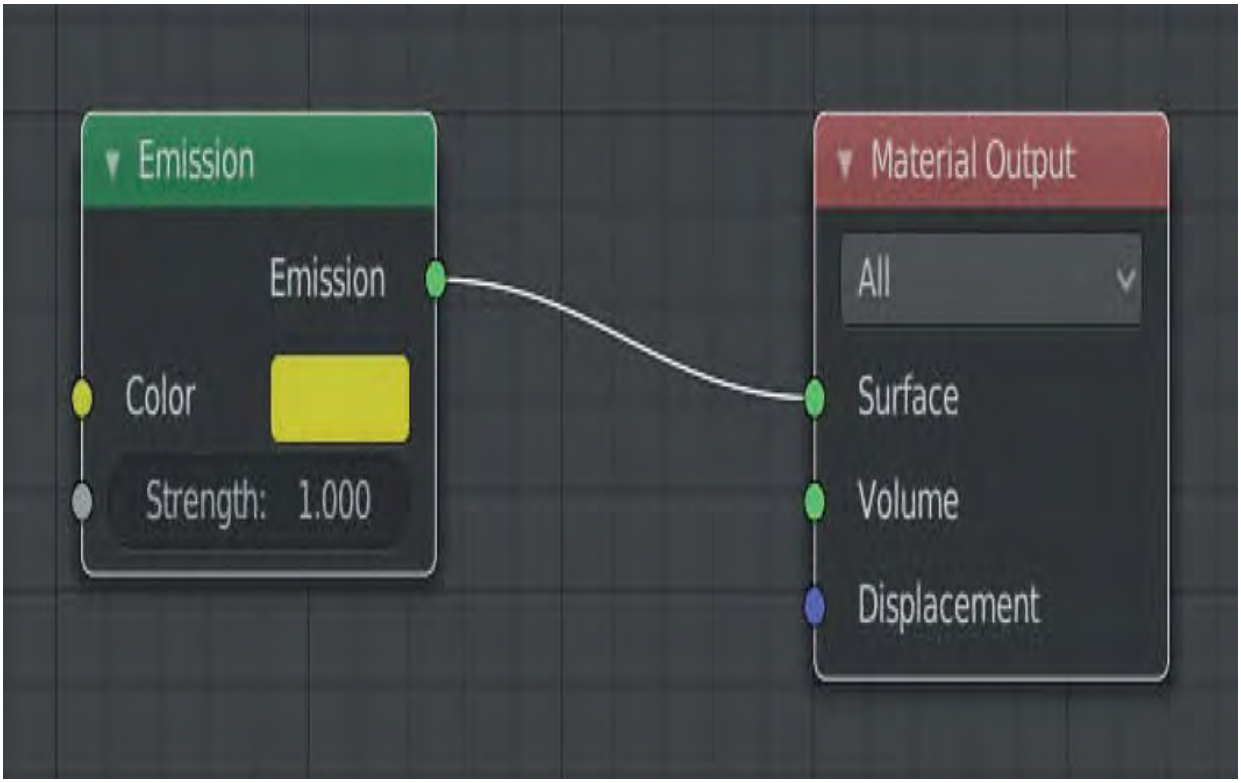


Figure 31.11

Figure 31.11 shows that in selecting Surface:Emission the Principled BSDF Node has been replaced by an Emission Shader Node in the Shader Editor.

Note: Bear in mind that **The Node system in Cycles is different from that used with Eevee Render.**

If you care to compare the Nodes available in the Add Menu you will find Cycles has a larger selection.

To demonstrate the Node system, start a new Blender file and place the **3D Viewport** in **Cycles Render** mode with **Viewport Shading – Rendered**. Change the **Timeline Editor** to the **Shader Editor**. Click on the **Material Properties** in the **Properties Editor**.

At this point you will have the gray Cube in the 3D Viewport Editor and the Principled BSDF Node connected to a Material Output Node in the Shader Editor.

The Shader Editor Header has Material Nodes activated by default with Use Nodes checked.

Note: You are working with the **Material Properties** in the **Properties Editor,** therefore, the **Material Node button** in the **Shader Editor** is active.

The Nodes displayed are a Principled BSDF Node and a Material Output Node (Figure 31.12).

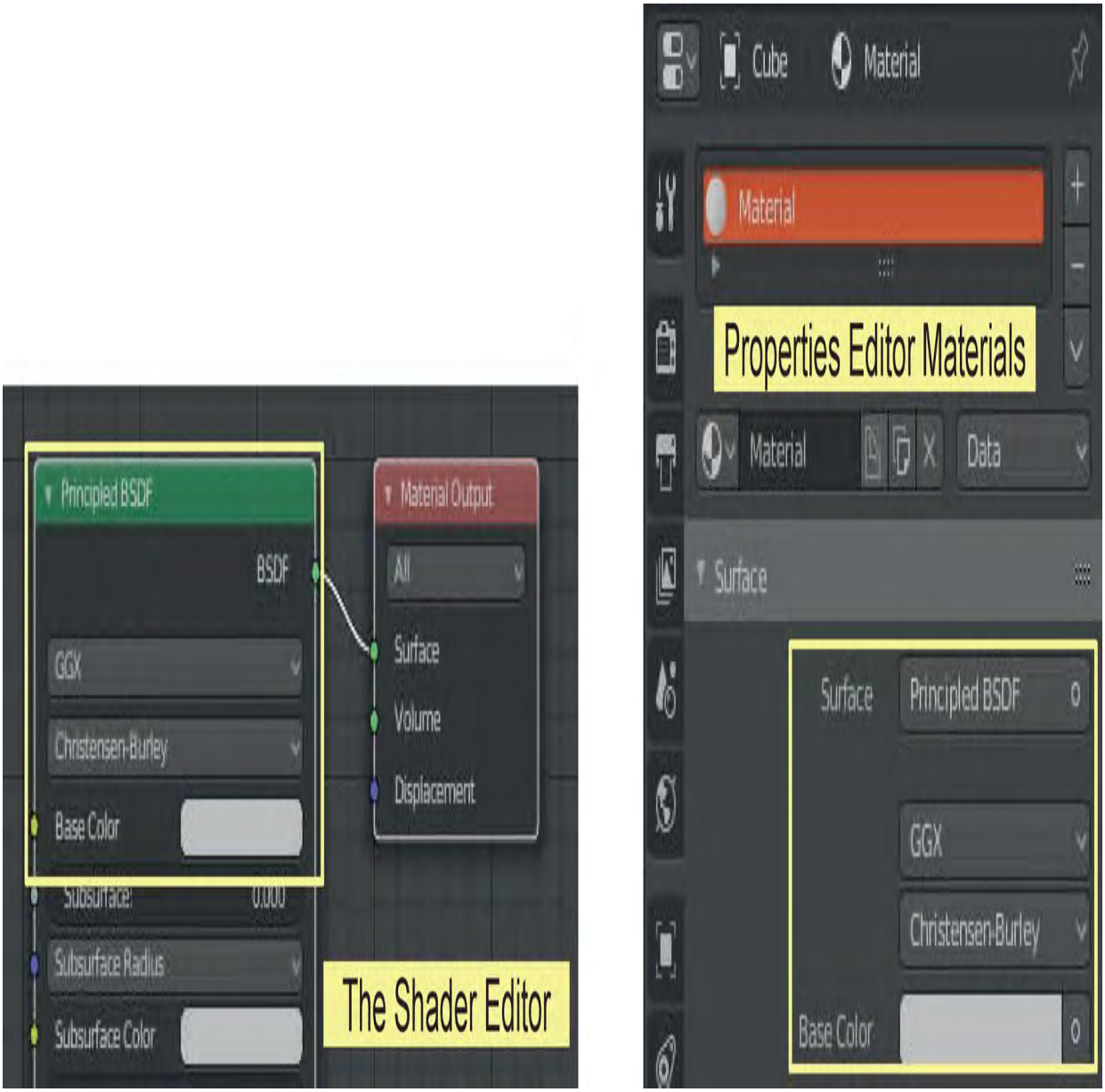


Figure 31.12

The Principled BSDF Node is a replica of the **Surface Tab** in the Properties Editor and note, this node is connected to the **Surface** input socket on the **Material Output** Node. In other words the Base Color is being mapped to the surface of the selected Object.

Clicking on the Base Color bar in the **Diffuse BSDF Node** displays a color picker where a new color may be selected, the same as the Base Color bar in

the **Properties Editor** (Figure 31.12).

31.4 Cycles in Practise

To demonstrate the practical application of **Cycles Render**, work through the following demonstration which will show you how to create and illuminate a Scene.

Set up the Scene

Arrange a UV Sphere and a Monkey **inside** the default Cube in a new Blender Scene as shown (Figure 31.13 and 31.14). The UV Sphere and the Monkey have smooth shading applied and the Cube has been scaled up and elongated and subdivided with several faces deleted on the back side creating a window. A **Solidify Modifier** has been applied to the modified Cube to create wall thickness.

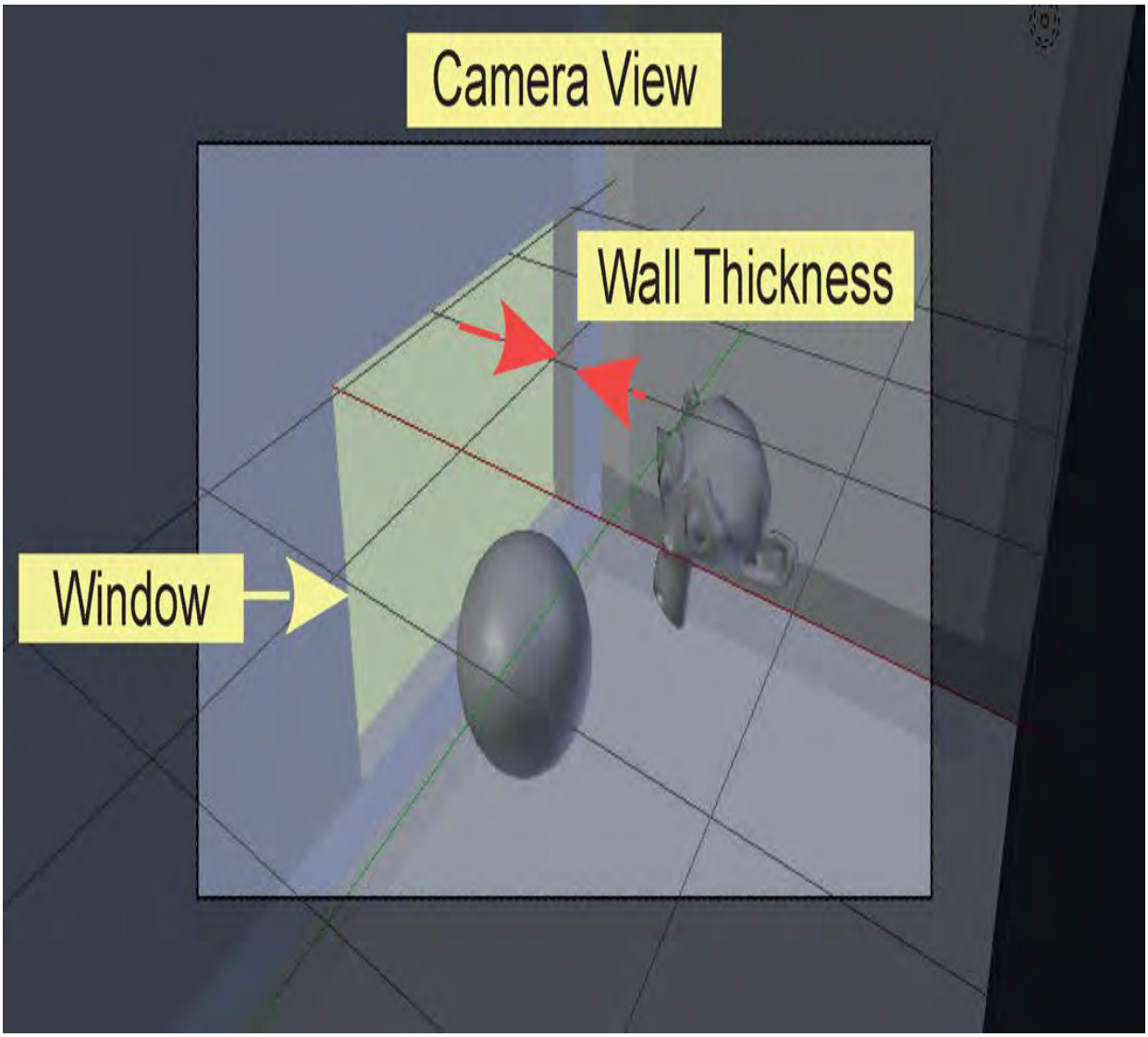


Figure 31.13

The objective is to place the Sphere and Monkey inside a room next to a window.

The Camera and single Point Lamp are also inside the room.

The Sphere, the Monkey and the window are positioned in Camera View.

At this stage the scene is illuminated by the default point lamp which is inside the room. Pressing F12 renders (Cycles Render – Rendered Viewport Shading) an image of the Camera view in Figure 31.15. The Render may take a while.

With the 3D Viewport Editor in **Solid Viewport Shading** mode all you see is the Cube. To see Objects inside the room change the 3D Viewport Editor to **Wireframe Viewport Shading**.

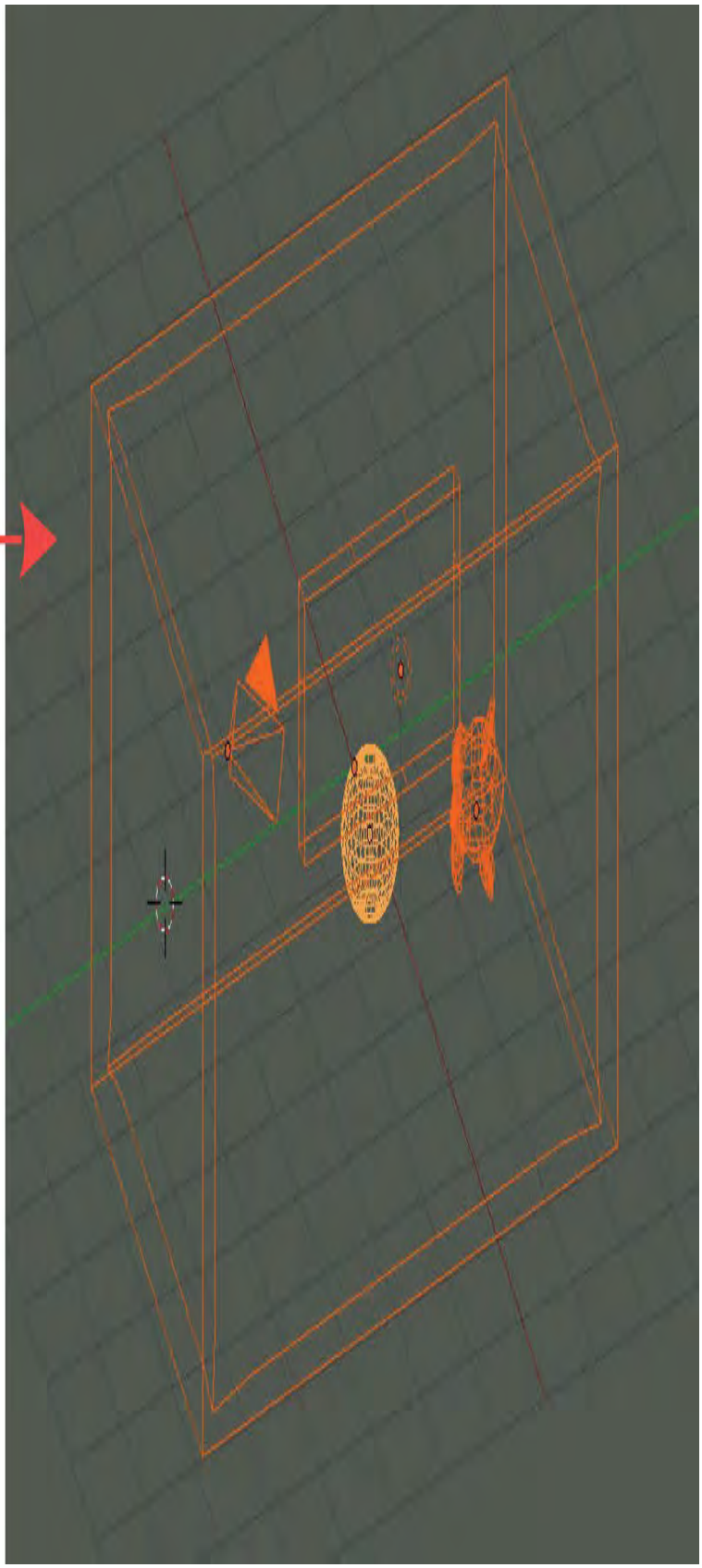


Figure 31.14

More Set Up Stuff

You may hide an Object temporarily to unclutter a Scene. In Solid Viewport Shading mode select the Cube (the room). Press the **H key**. To reinstate the Cube press **Alt + H key**. This is not particularly useful in this instance but is worth remembering.

In the Properties Editor, Render button, change **Eevee Render** to **Cycles Render.** Have the 3D Viewport Editor in Camera Perspective View and change to Rendered Viewport Shading Mode. This will allow you to see rendered results as you progress.

If you have GPU rendering available and it is faster than CPU rendering it's time to turn it on.

> **Warning:** Turning GPU may or may not be beneficial depending on your PC. Rendering in Cycles may stall or crash the program. You will have to experiment.

You may go to the **Properties window, Render Properties, Sampling tab** and change the Preview Samples value. The default is 32 which doesn't allow a fantastic result (Figure 31.15). Remember: Increasing the value will increase the render time.

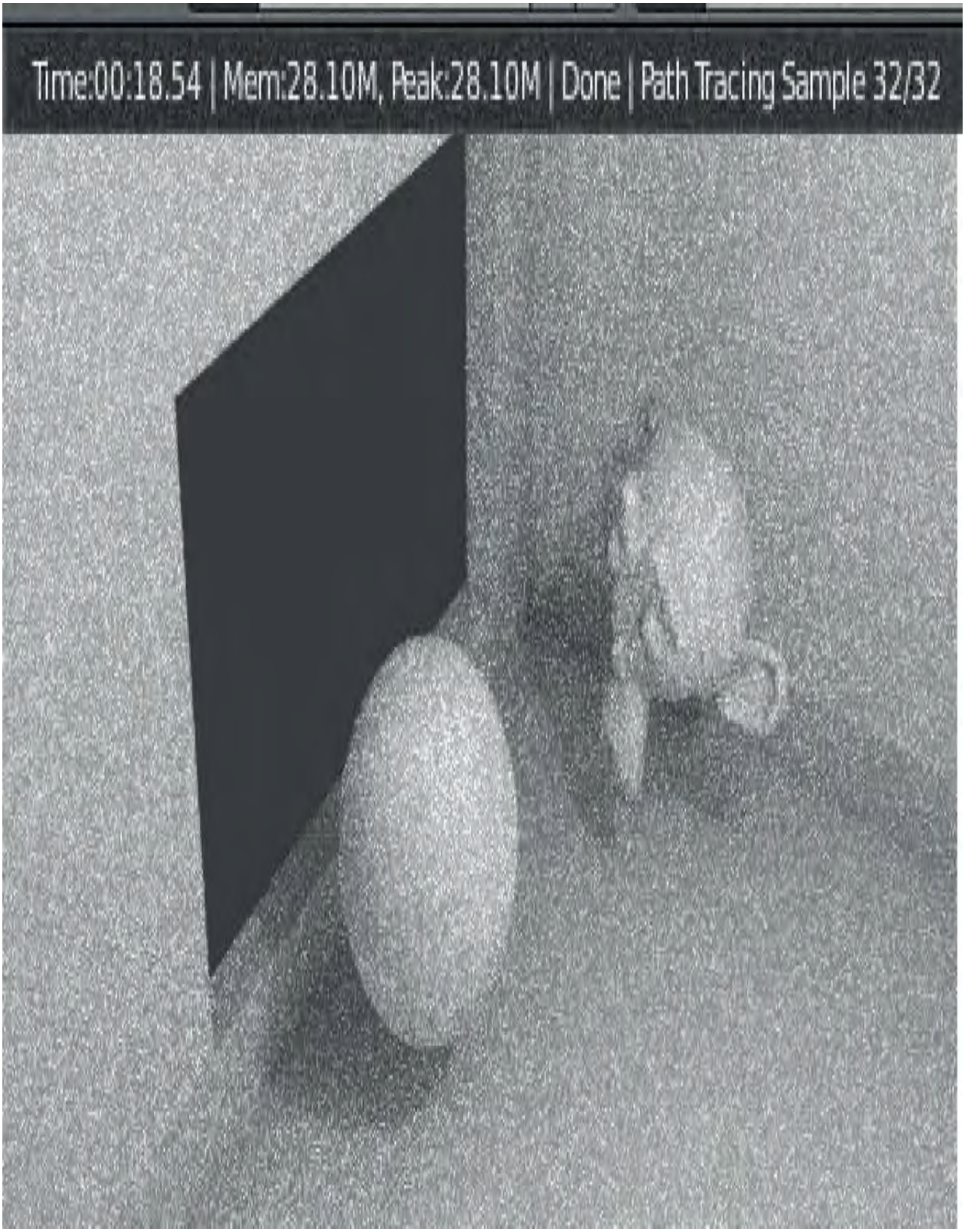

Time:00:18.54 | Mem:28.10M, Peak:28.10M | Done | Path Tracing Sample 32/32

Figure 31.15

By comparison a full render (F12) at 128 Samples is shown in Figure 31.16.

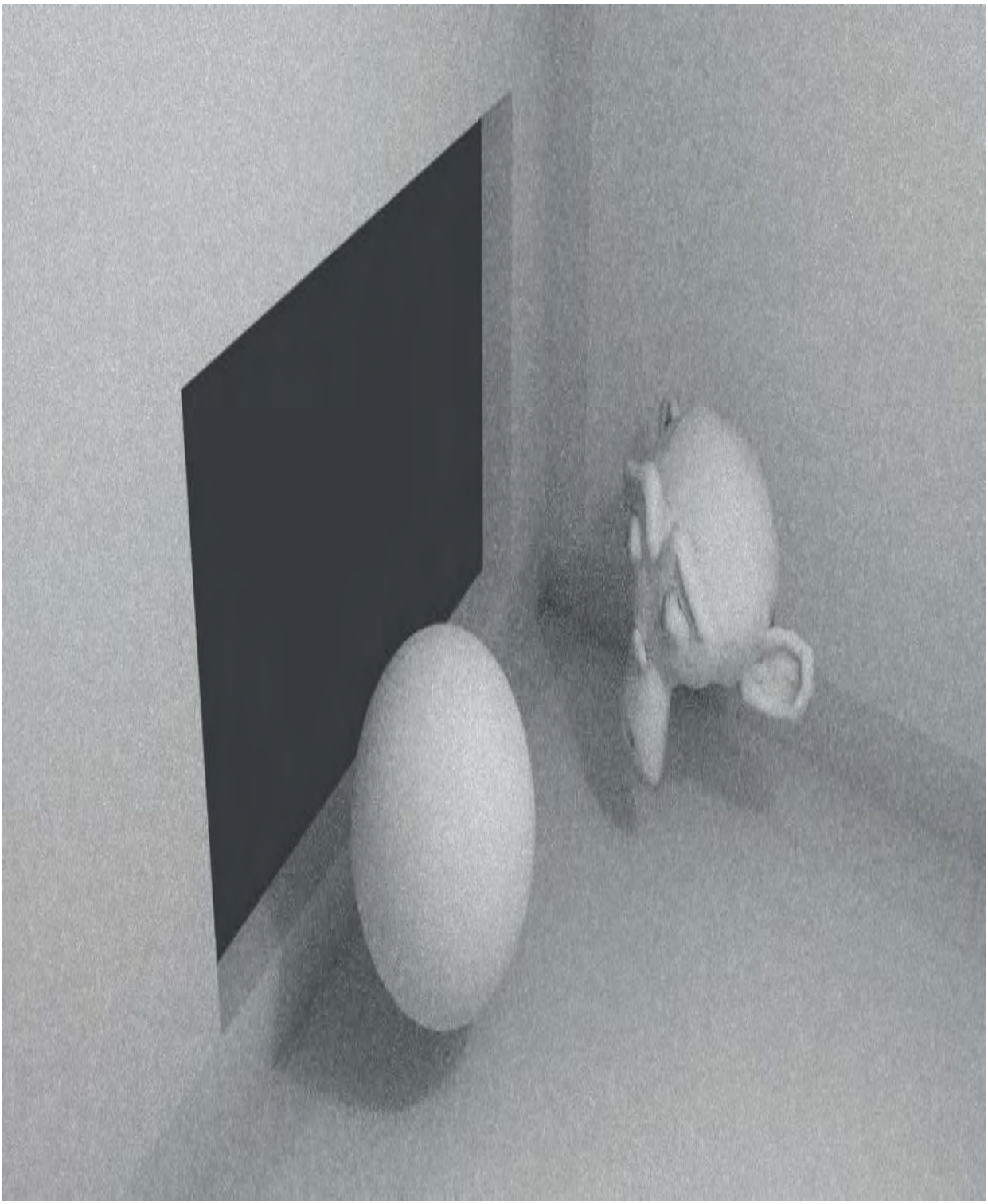


Figure 31.16

Materials in the Properties Editor

Select the Monkey in the 3D View Editor. Remember: The monkey is named **Suzanne** as you can see in the upper left hand corner of the **3D View Editor** and in the **Outliner Editor**.

In the **Properties Editor, Material buttons**, click on the **New** button. As previously explained the Material Nodes for Cycles Render are different to Eevee Render.

The default settings show the **Principled BSDF** Node active. The Color (the white color bar/picker) is showing Blender's default gray, RGB 0.800. Click on the **Surface** button where you see Principled BSDF and select **Glass BSDF (**Figure 31.17).

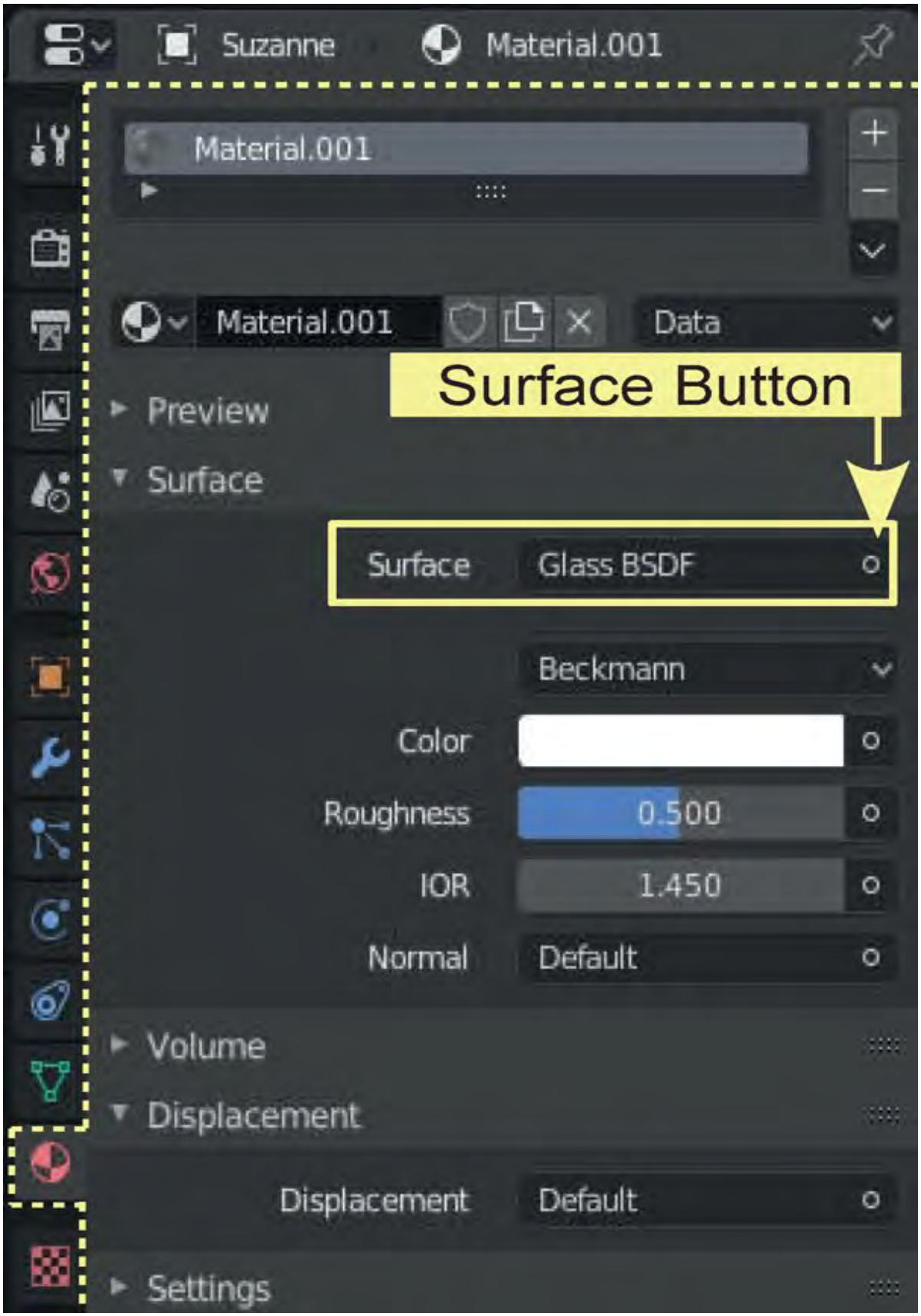

Suzanne
Material.001
Material.001
Material.001
Data
Surface Button
Preview
Surface
Surface
Glass BSDF
Beckmann
Color
Roughness
0.500
IOR
1.450
Normal
Default
Volume
Displacement
Displacement
Default
Settings

Figure 31.17

This gives Suzanne a gray glass like effect. Change the color and reduce the Roughness value to 0.000. The effects of the Surface Shader and color are immediately seen in the Camera view (Figure 31.18 – **Note:** Render Sample 200 and Time taken to complete the render 2 minutes, 8.68 seconds).

You will have to decide if the wait is worth it.

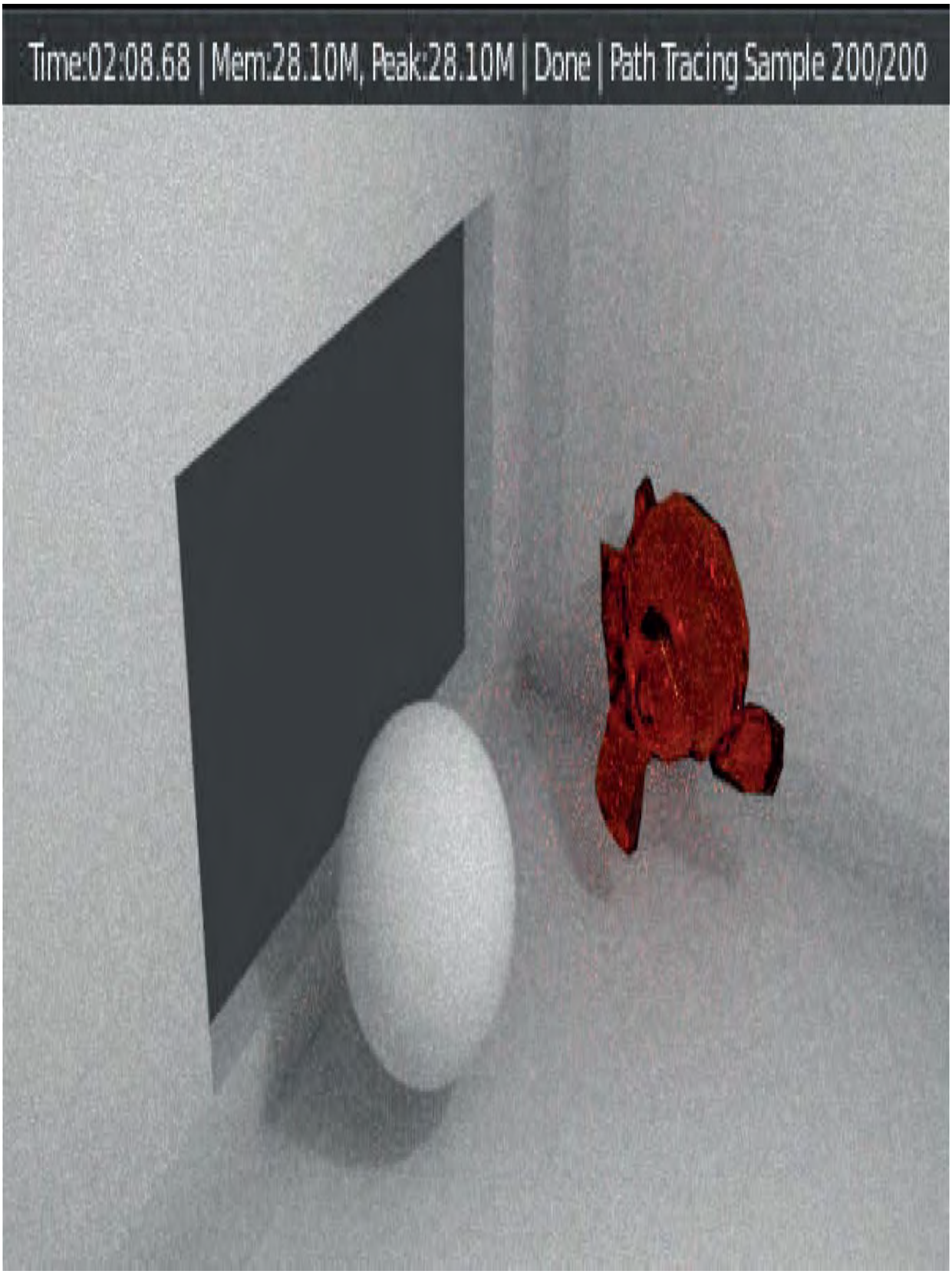

Time:02:08.68 | Mem:28.10M, Peak:28.10M | Done | Path Tracing Sample 200/200

Figure 31.18

This procedure has applied a **Material** to the Monkey by using the **Properties Editor, Materials buttons**.

Remember: The application of Materials using Cycles Render is using the **Cycles Node System**. You may use the Properties Editor or the Shader Editor to apply Materials.

Note: Adding, manipulating and connecting nodes when using **Cycles Render** is the same as when using Eevee Render. The difference is, in Cycles Render there are more Node options to choose.

Note: Node Groups created when using **Eevee** and **Appended** into a file using **Cycles** will not work unless have been to do so.

Materials in the Shader Editor

Deselect Suzanne and select the **UV Sphere**. Change the Timeline Editor at the bottom of the Screen to the Shader Editor and drag the top edge up. Place the 3D View Editor at the top of the Screen in Rendered Viewport Shading mode.

In the Rendered View the UV Sphere displays with Blender's default gray Material color even though a Material has not been applied. With the UV Sphere selected the Shader Editor is empty. By default Blender applies the default gray color to the UV Sphere. Click on the New button in the Shader Editor Header to activate the Node System. In the Shader Editor the Principled BSDF Node displays connected to the Material Output Node. The Properties Editor, Material buttons show Principled BSDF in the Surface Tab.

With the Mouse Cursor in the **Shader Editor** click LMB to deselect all Nodes then LMB click on the Principled BSDF Node to select. Press the X Key to delete the Principled BSDF Node. The UV Sphere displays black in the 3D View Editor.

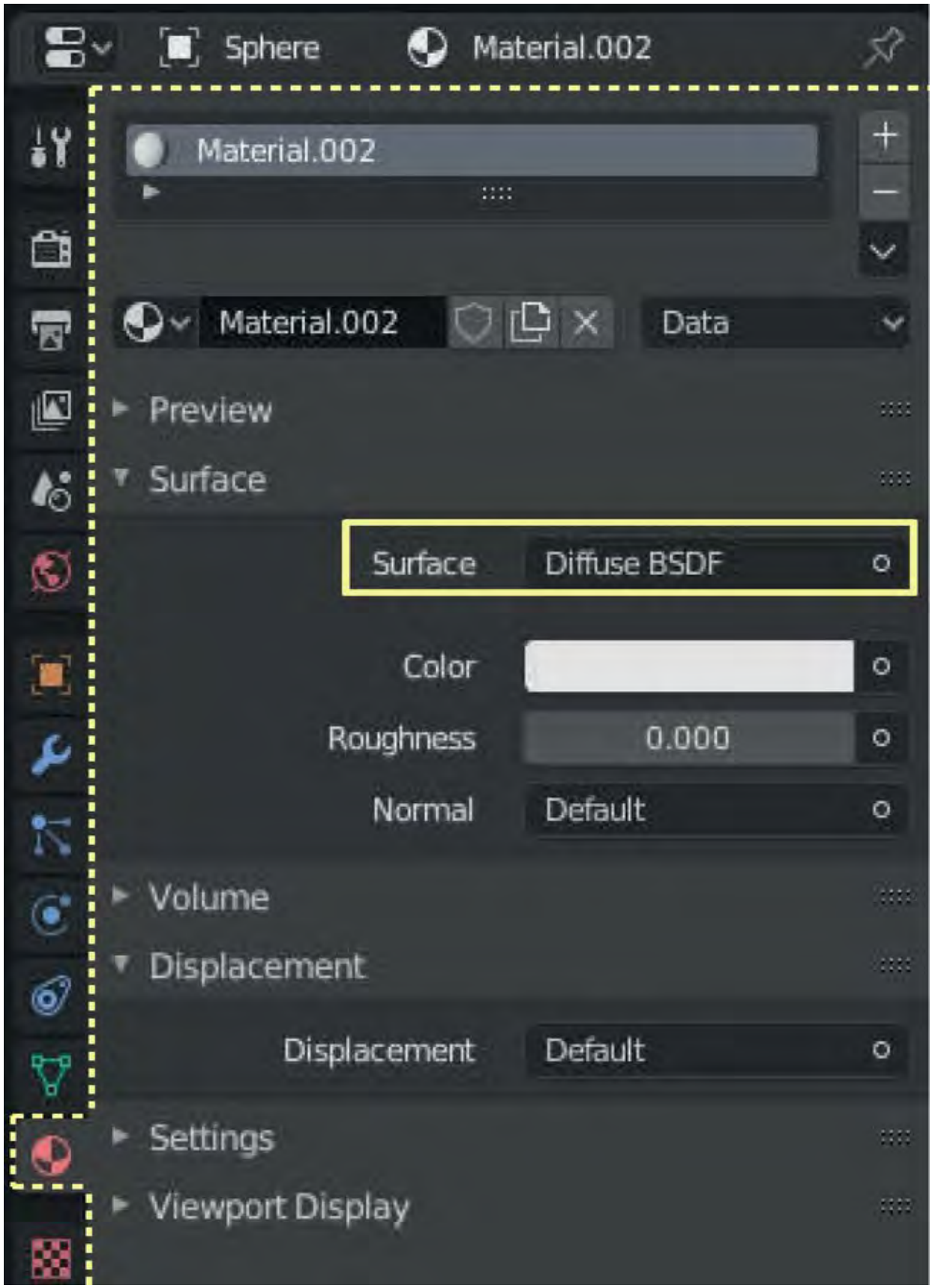

Sphere
Material.002
Material.002
Material.002
Data
Preview
Surface
Surface
Diffuse BSDF
Color
Roughness
0.000
Normal
Default
Volume
Displacement
Displacement
Default
Settings
Viewport Display

Figure 31.19

Click Add in the Shader Editor Header and select Shader – Diffuse BSDF to enter a Node. Connect the Diffuse BSDF Node output socket to the Surface input socket of the Material Output Node.

In the **Shader Editor** the **Diffuse BSDF**. This Node is the graphical representation of the data in the **Properties Editor**. The **Material Output** Node is the graphical representation of data sending the Diffuse material color to the surface of the UV Sphere.

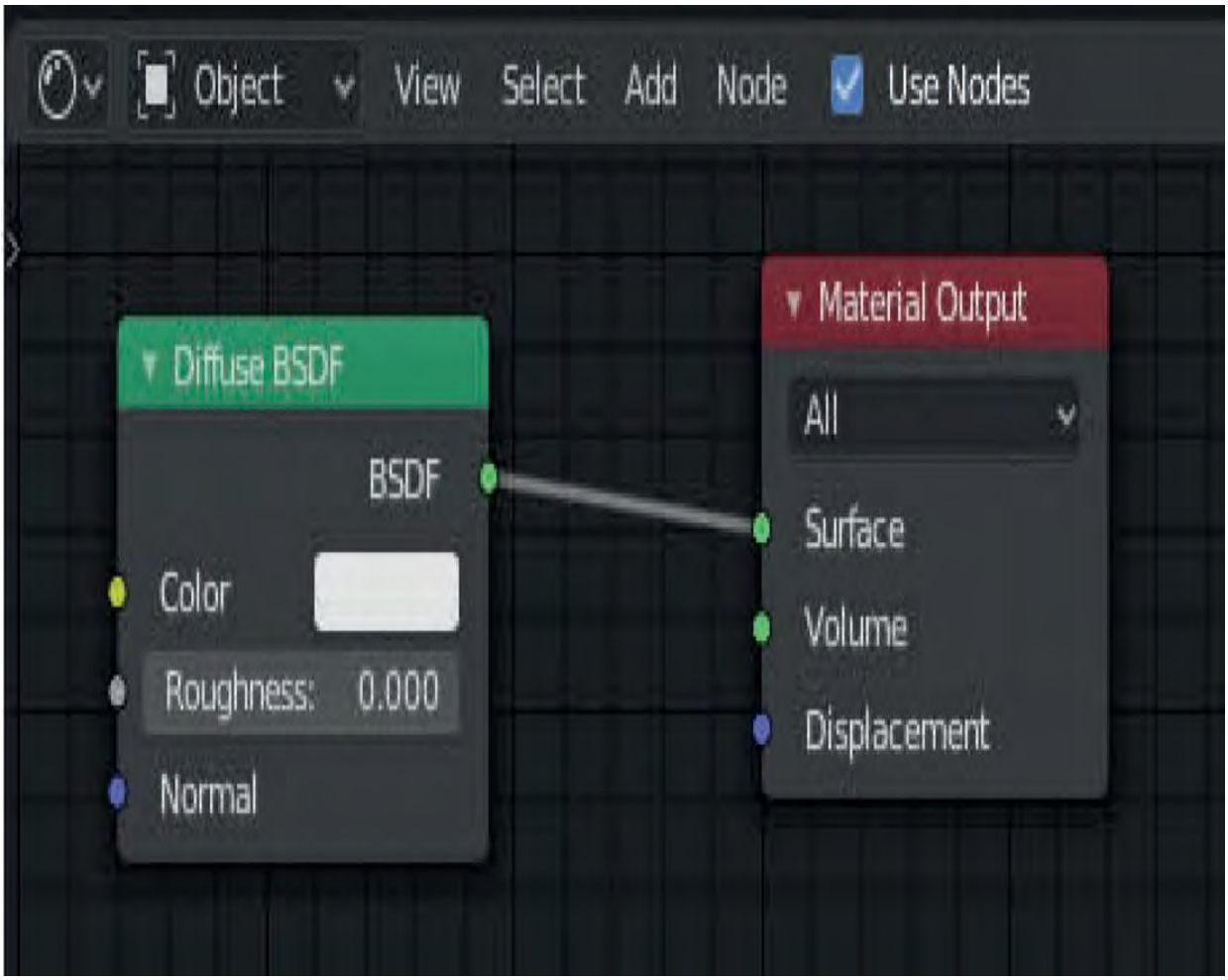


Figure 31.20

The display in the 3D Viewport Editor, Rendered Viewport Shading, Camera view is as shown previously in Figure 31.15.

The Power of Nodes

To discover the power of using Nodes in Cycles Render, have a UV Sphere positioned above a Plane in a new blender Scene (inset Figure 31.2**1**) and perform the following:

With the UV Sphere selected add a Glossy BSDF Node (click Add in the header – Shader -Glossy BSDF). Add a Mix Shader Node (click Add in the header – Shader – Mix Shader). Arrange and connect the Nodes as shown in Figure 31.2**1**.

In the Glossy BSDF Node click on the Color button to display the color picker and select a soft pastel color. The UV Sphere in the 3D Viewport Editor will immediately render showing a glossy surface.

In the Diffuse BSDF Node select a bright vivid color. The 3D Viewport Editor re-renders, mixing the two Shaders together. The Roughness sliders in the Shader Nodes vary the intensity of the colors while the Fac: slider in the Mix Shader Node adjusts the ratio of the mix.

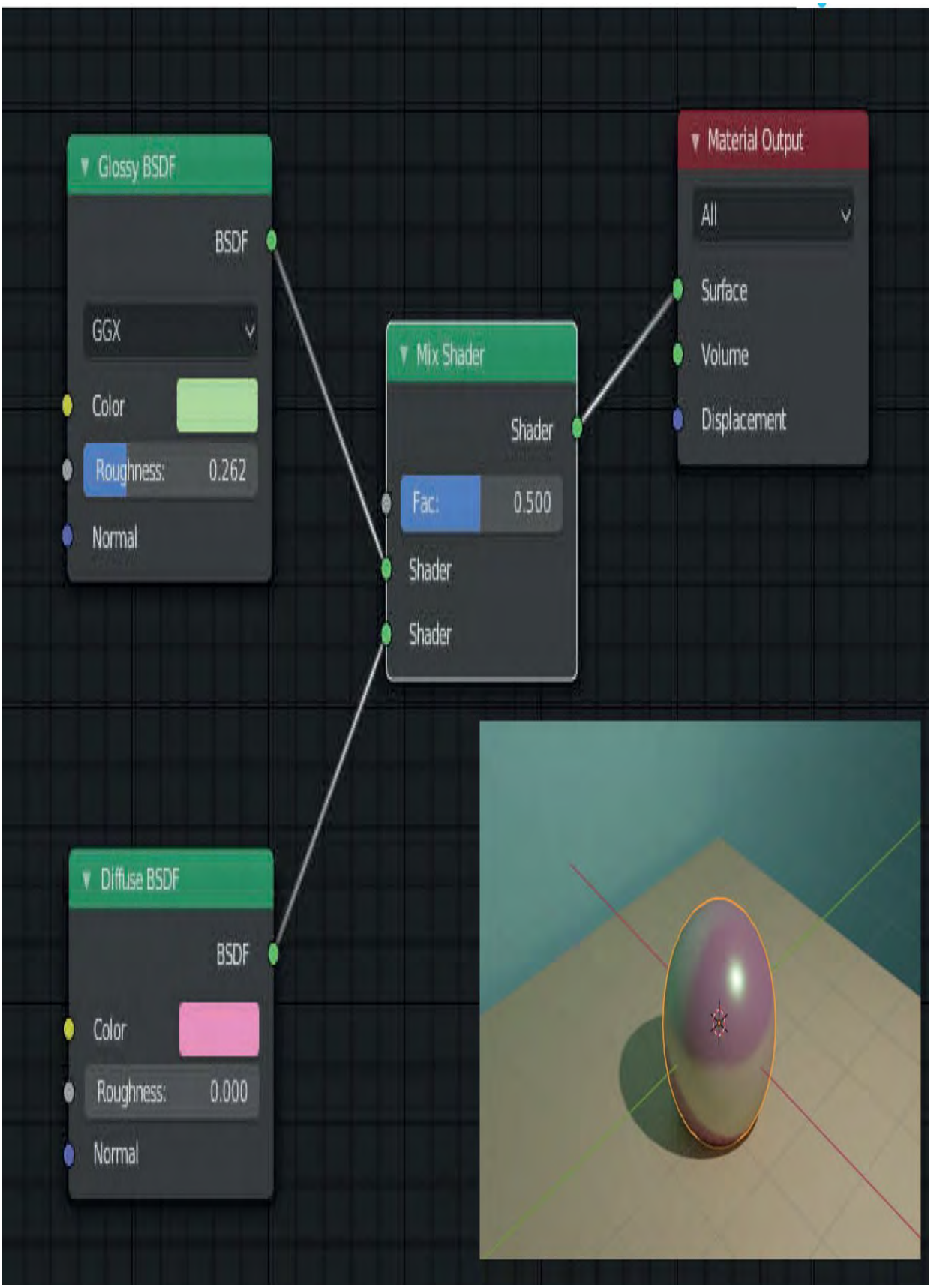

Glossy BSDF
BSDF
GGX
Color
Roughness: 0.262
Normal
Material Output
All
Surface
Volume
Displacement
Mix Shader
Shader
Fac: 0.500
Shader
Shader
Diffuse BSDF
BSDF
Color
Roughness: 0.000
Normal

Figure 31.21

There are numerous combinations of Nodes available and it is impossible to cover everything but this simple demonstration shows the potential of one aspect of Nodes in Cycles.

What you see in the rendered view has been reliant on the default single point lamp for illumination. This can be vastly improved in Cycles.

Sky Lighting

The effect of Sky Lighting is to create an ambient light which is the overall lighting from the sky in the natural world. In the Scene inside the room the Camera is positioned inside. Camera View would only show this through the window.

To have a better understanding of the effect, use the Scene with the UV Sphere above the Plane.

In the **Properties Editor, World buttons, Surface tab**, click at the end of the **Color bar**, click on the button and select **Sky Texture** from the menu (Figure 31.23). Voila! you have blue sky in the window background. In the **Color bar** you will see **Sky Texture**.

Click, hold and drag the mouse on the **Sky Ball** to adjust shadow effect in the Scene. You may also adjust the **Turbidity** value between 1.00 and 10.00 for a variation. The Strength value also changes the sky effect. Sky Lighting sets the lighting for outdoors.

Figure 31.22 shows Sky Lighting with the values in Figure 31.23.

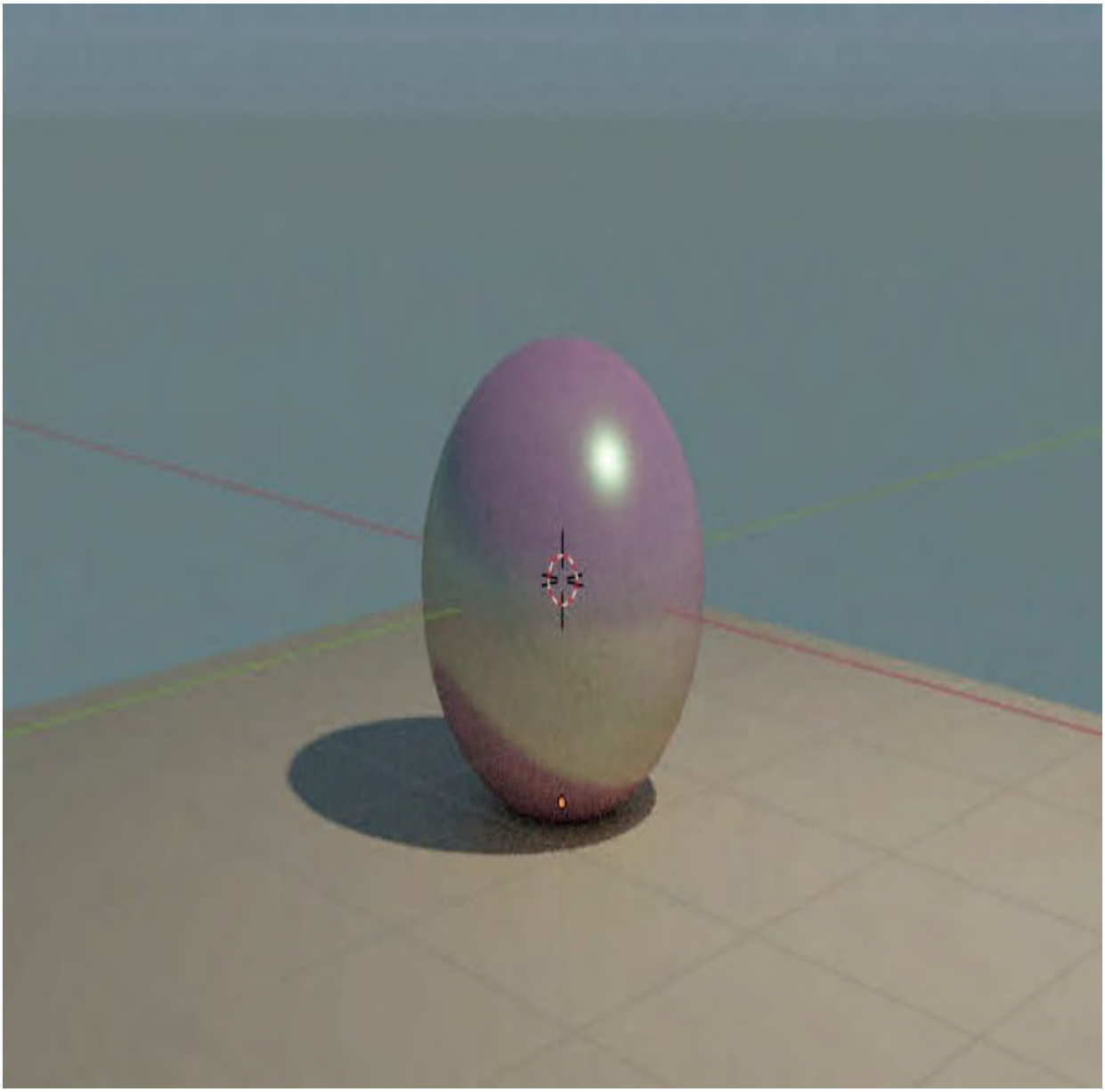


Figure 31.22

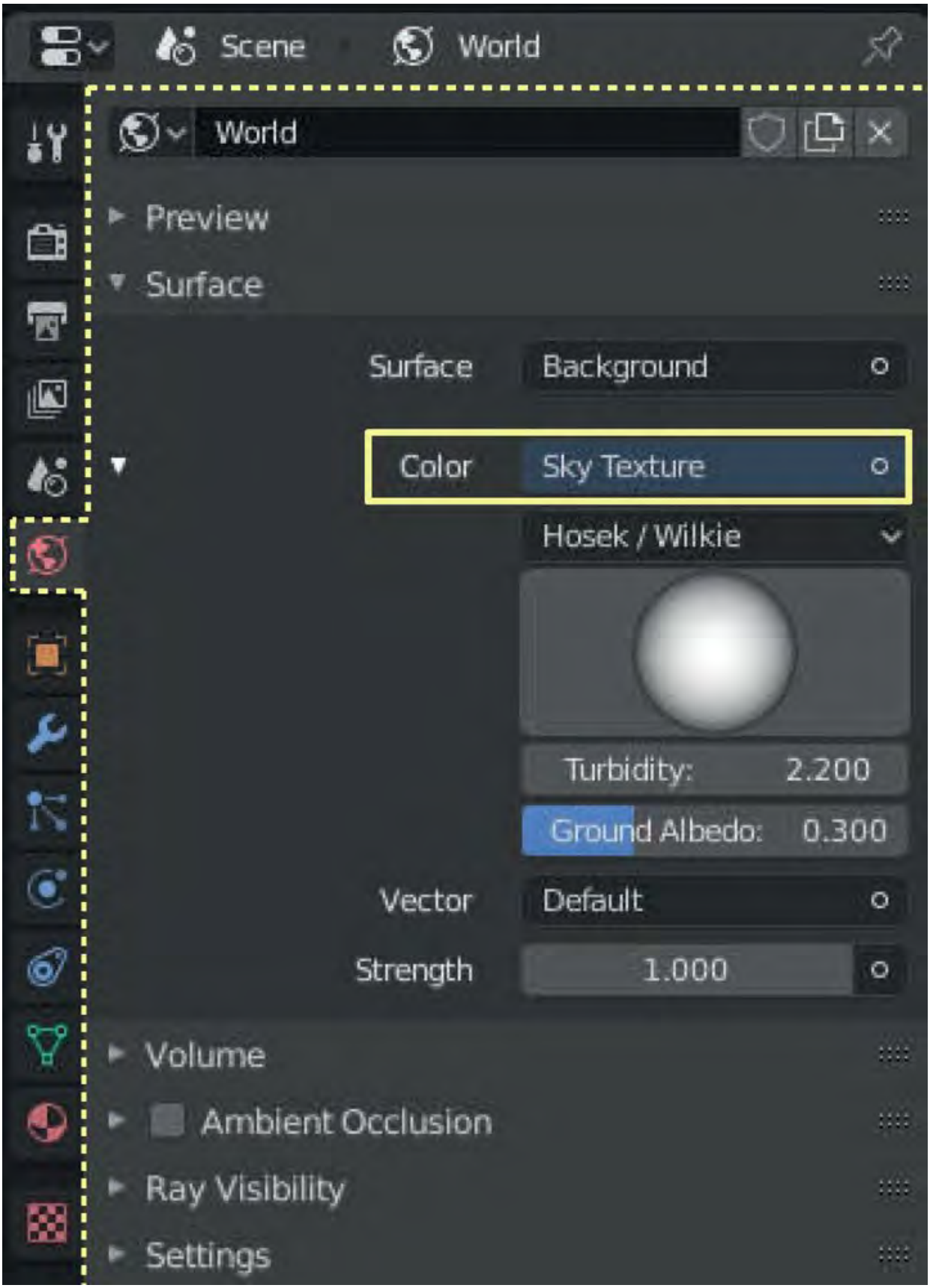

Scene
World
World
Preview
Surface
Surface
Background
Color
Sky Texture
Hosek / Wilkie
Turbidity:
2.200
Ground Albedo:
0.300
Vector
Default
Strength
1.000
Volume
Ambient Occlusion
Ray Visibility
Settings

Figure 31.23

You see the Node Arrangement for the Sky effect produced by the values set in Figure 31.23 in the Shader Editor with World Shader Nodes active (Figure 31.24 – Editor Header only).

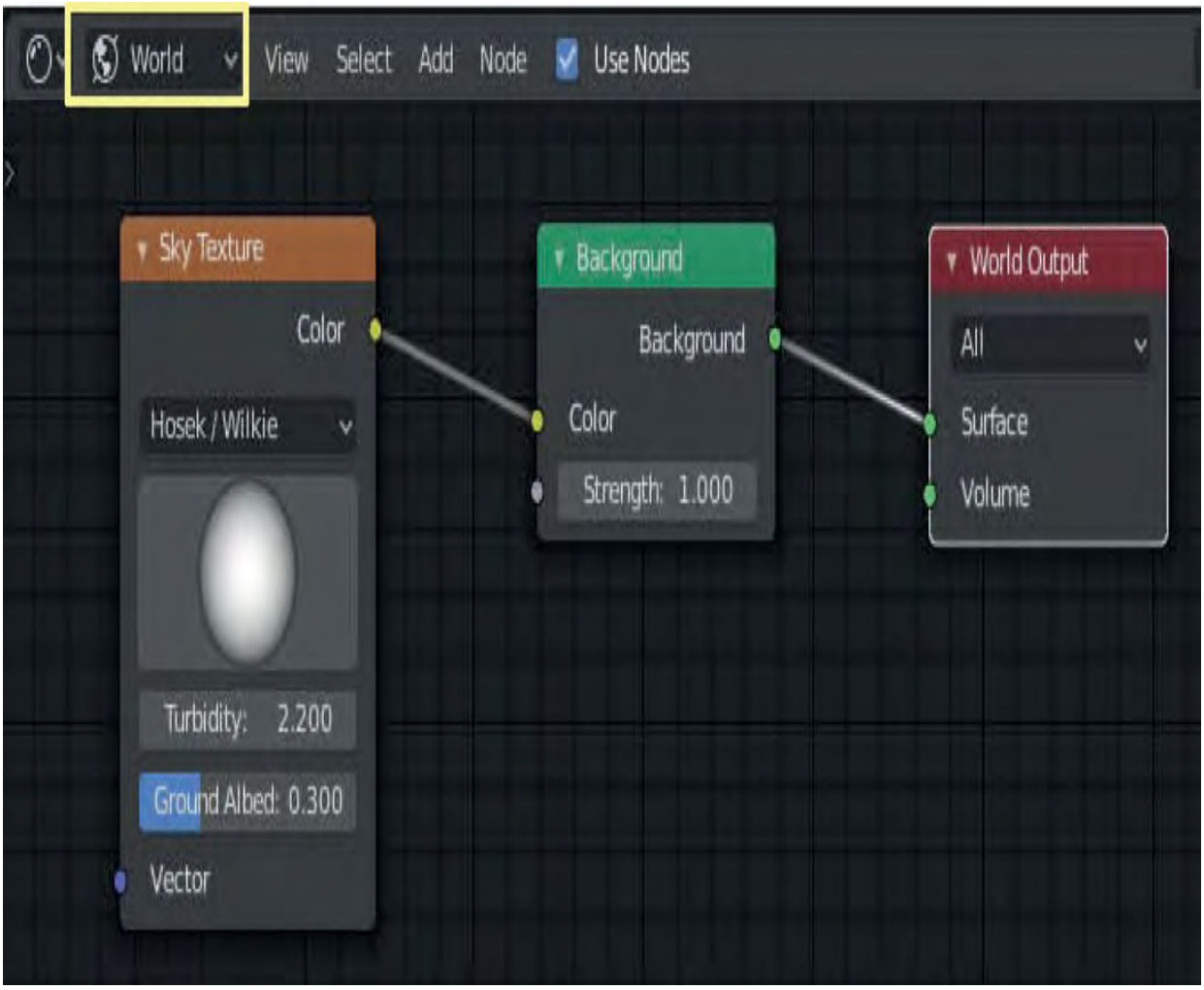


Figure 31.24

Its time to go back indoors.

Reinstate the Room:

With Sky Lighting created delete the Point Lamp. There will be no light inside the room but light will come through the window from outside (Figure 31.25).

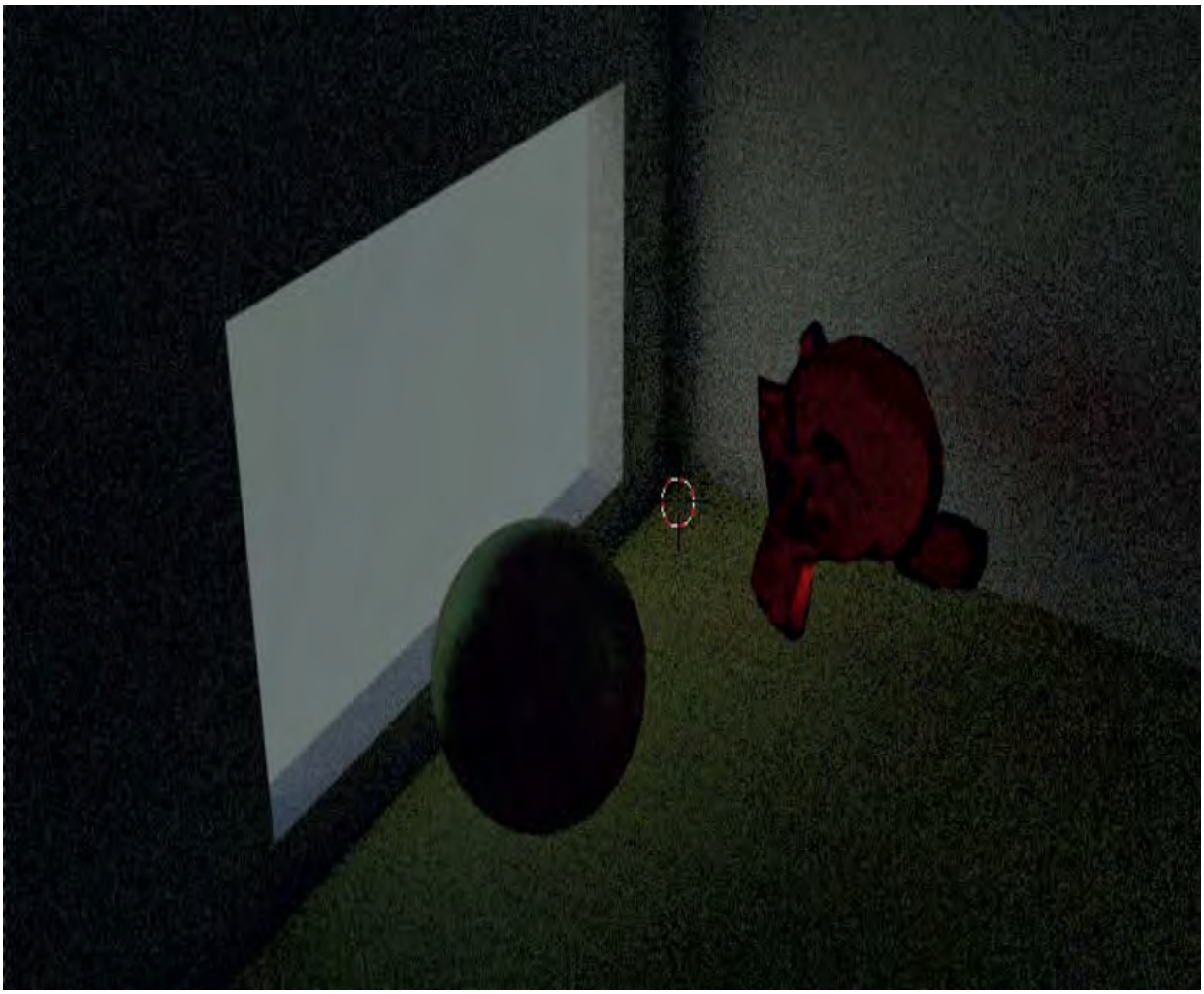


Figure 31.25

Object Lighting

The Point Lamp will be replaced by an Object as a source.

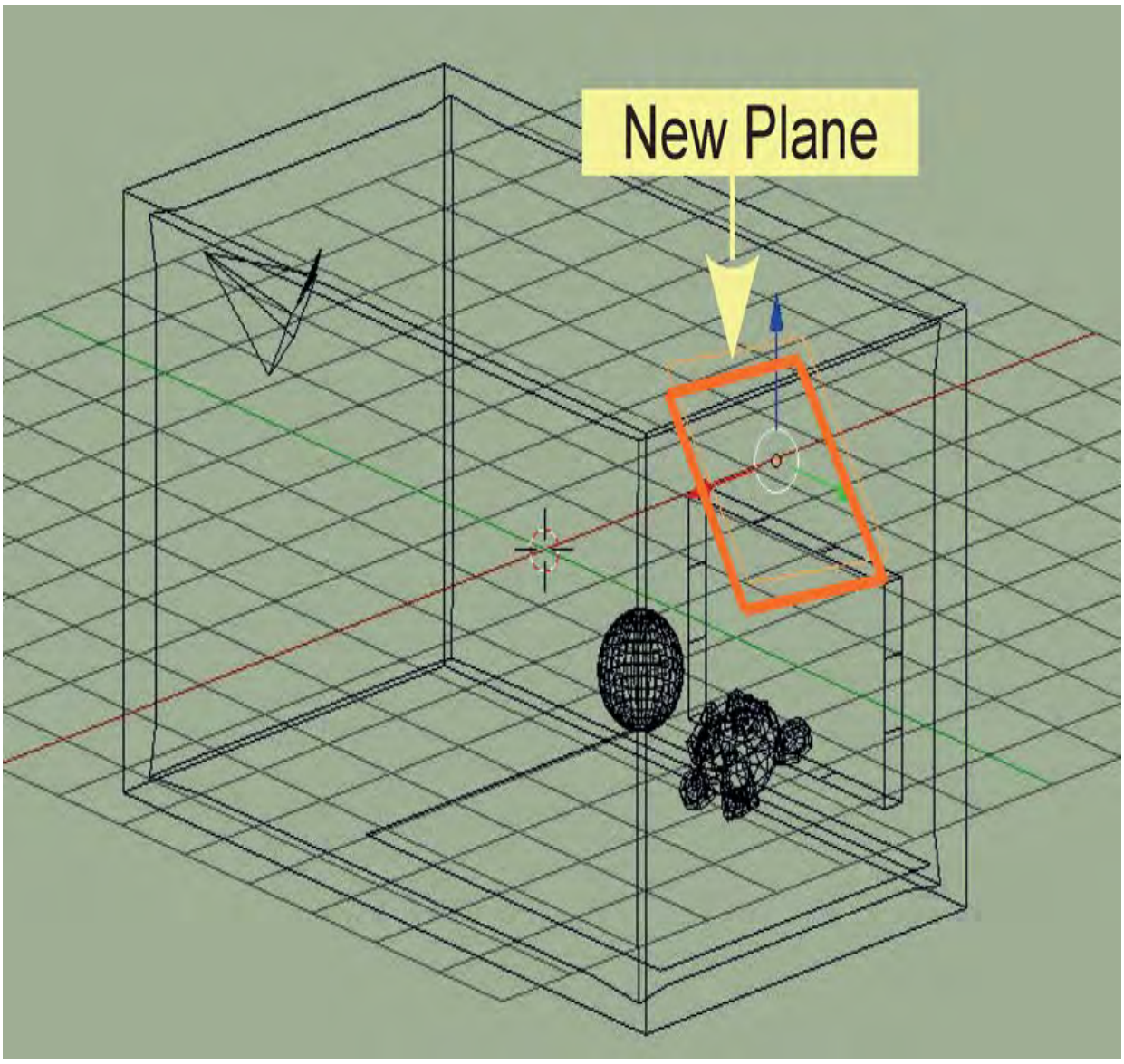


Figure 31.26

Add a Plane and position as shown in Figure 31.26. You want the Plane above and angled towards the Objects. Think of it as a mirror which will reflect light onto the Objects.

Make sure the Plane, when in Camera view, is not in the aperture.

In the **Properties Editor, Material buttons**, click **New** to add a Material. Select **Surface, Shader** type, **Emission**. This makes the plane a light source (Figure 31.27).

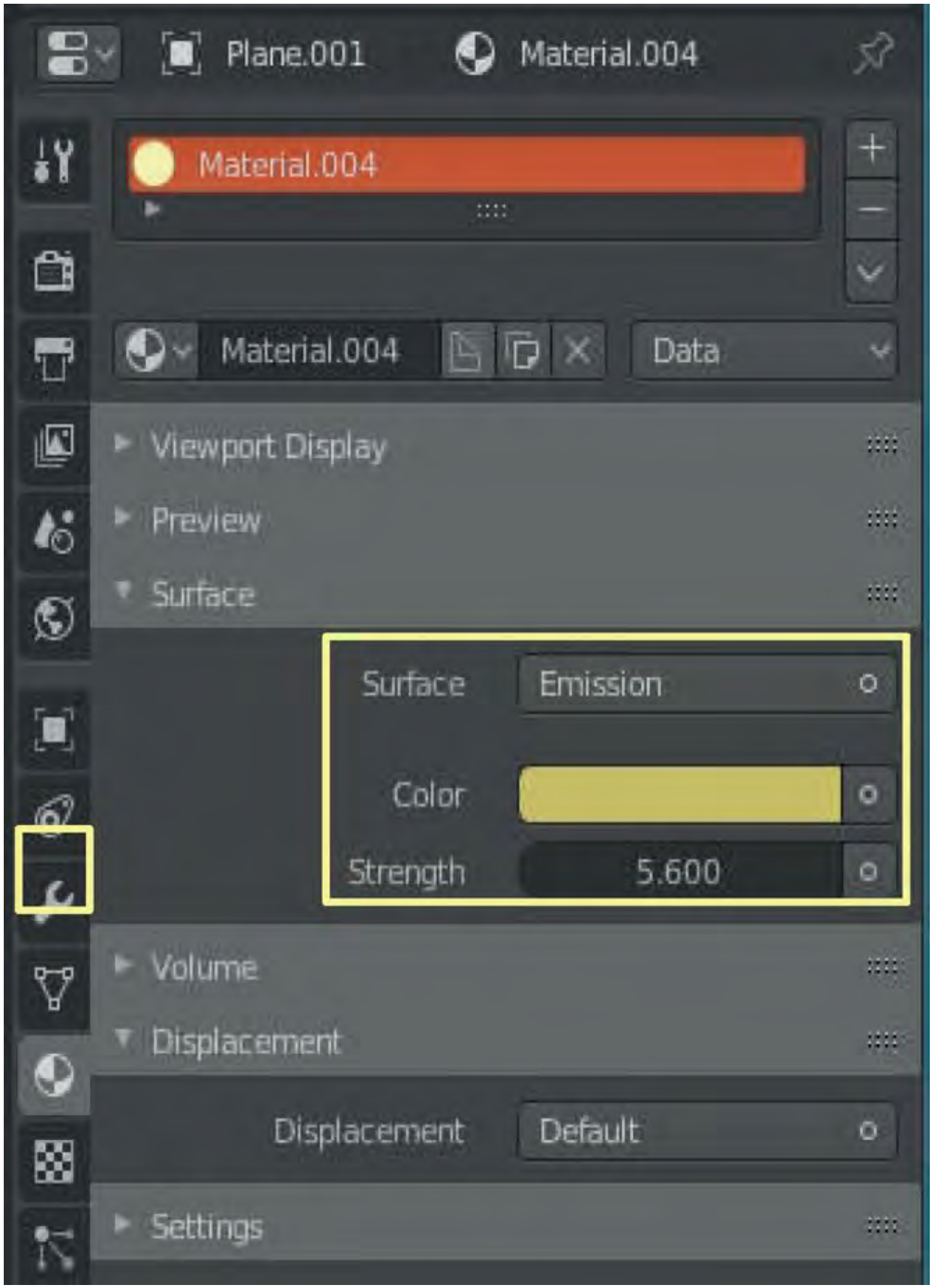

Plane.001
Material.004
Material.004
Material.004
Data
Viewport Display
Preview
Surface
Surface
Emission
Color
Strength
5.600
Volume
Displacement
Displacement
Default
Settings

Figure 31.27

With the **Camera View** in **Rendered mode** you see the inside of the room illuminated (Figure 31.28). The light Color can be changed and the Strength value adjusted. The position and scale of the light source affects the rendered view. You may add more than one light source.

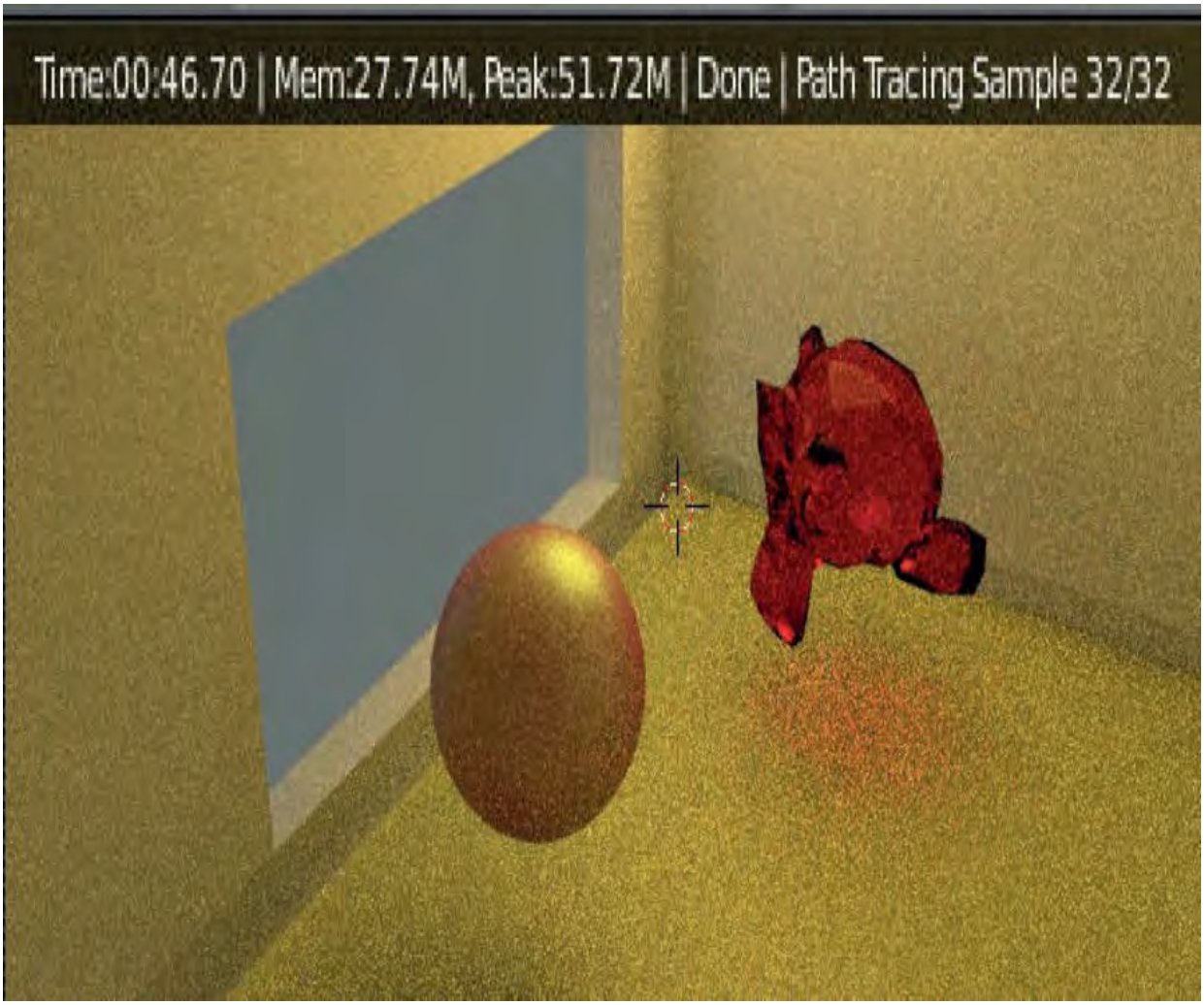


Figure 31.28

31.5 Workbench Render

This introduction to the **Workbench Render Engine** is intended to make you aware of its existence and to point out a few of its features. The following

description is taken from the Blender Wiki.

General Description

The Workbench Engine is a render engine optimized for fast rendering during modeling and animation preview. It is not intended to be a render engine that will render final images for a project. Its primary task is to display a scene in the 3D Viewport when it is being worked on.

Workbench also has an X-ray mode to see through objects, along with cavity and shadow shading to help display details in objects. Workbench supports several lighting mechanisms including studio lighting and MatCaps.

Lighting

The Workbench engine does not use the lights of the scene. The lighting conditions that will be used can be set in the Lighting panel (Tab).

Scene
Render Engine
Eevee
Click and select Workbench
64
Viewport
16
Viewport Denoising
Scene
Render Engine
Workbench
Sampling
Render
8 Samples
Viewport Render
Single Pass Anti-Aliasing
Lighting
Studio
MatCap
Flat
Rotation:
0°
Color
Material
Object
Vertex
Single
Random
Texture
Options
Film
Simplify
Color Management

Figure 31.29

Workbench Render is accessed in the Properties Editor, Render buttons (Figure 31.29).

With Workbench opened the Render buttons display with the Sampling Tab, the Lighting Tab and the Color Tab open. Settings in these tabs will directly control the display in the 3D Viewport Editor and supersede normal controls.

For example: With Workbench active the Viewport Display

Modes are limited.

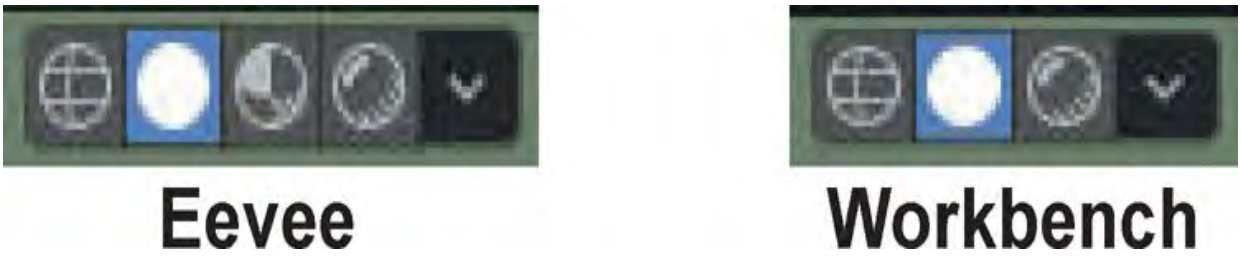


Figure 31.30

You will also find the Materials buttons in the Properties Editor limited or appear to be. More on this later.

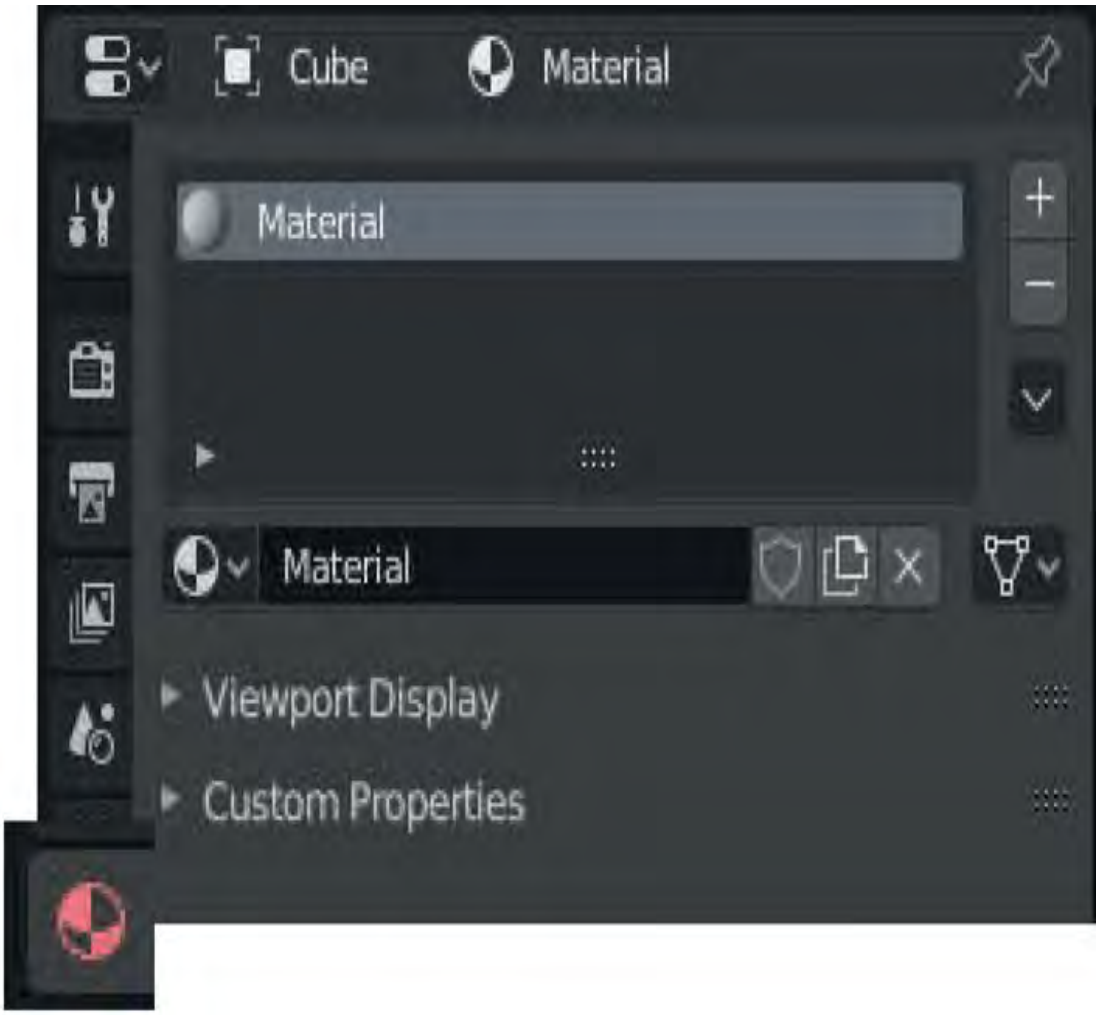


Figure 31.31

The display in the 3D Viewport Editor is controlled from the Properties Editor Render buttons. To use Materials as an example, a Material applied to the default Cube in the default Scene with Eevee Render active does not display in the Viewport when you switch to Workbench Render.

Workbench is designed to present a complicated Scene in a simplified display for modeling, and at the same time provide previews of color and lighting effects as previews of a final Render.

The Scene shown in Figure 31.32 is far from being complicated but it will serve to demonstrateWorkbench controls.

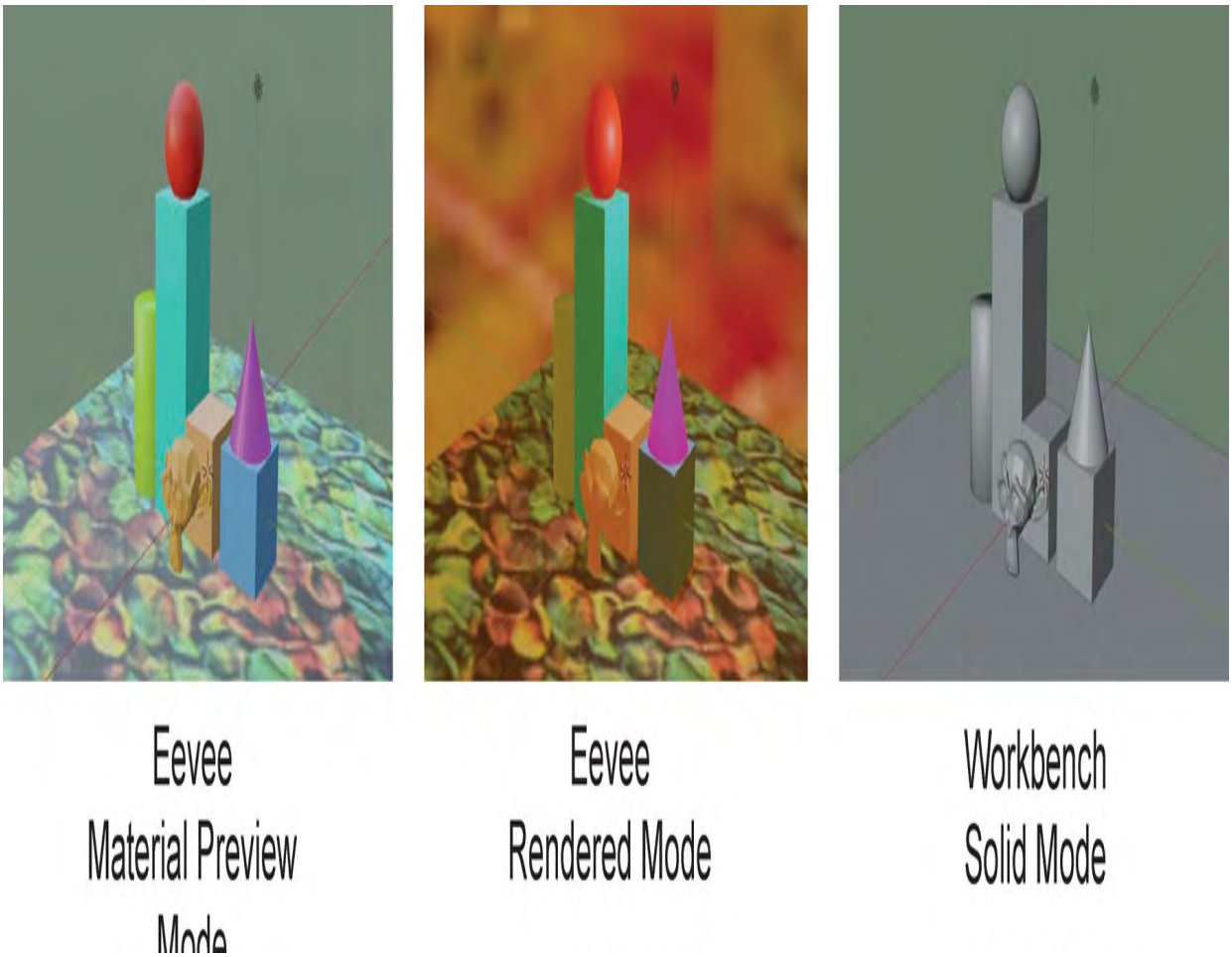


Figure 31.32

With Materials added to the Objects the colors display in Eevee Render in Material Preview and Rendered display Modes. Changing to Workbench Render sees the Objects in the default Gray. In Figure 31.32 the Workbench Solid Mode shows a green Screen background. Changing to Workbench Render with Rendered Viewport Shading shows a gray World background (Figure 31.33). This is the starting point for exploring the Properties Editor, Render buttons for Workbench Render.

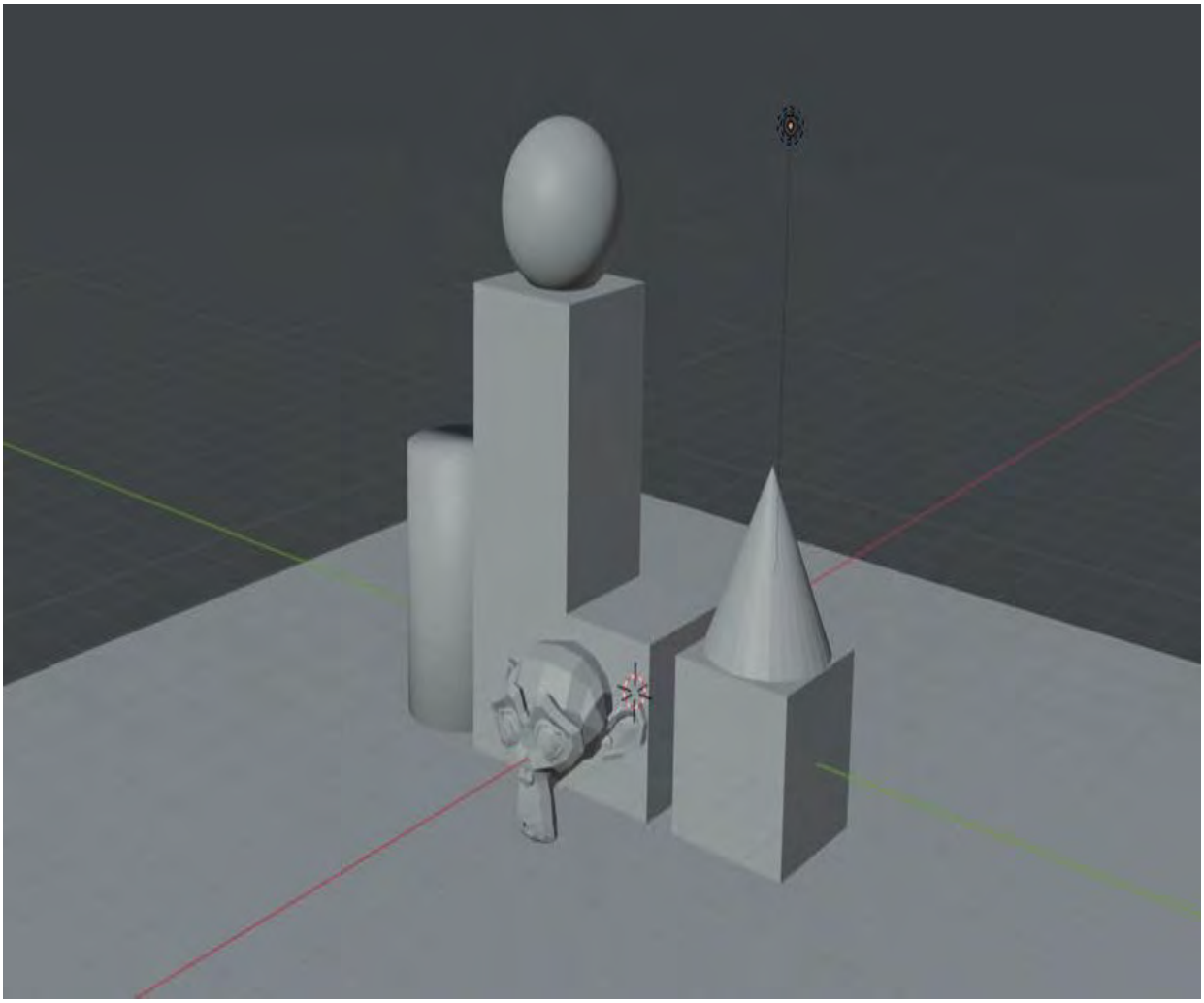


Figure 31.33

Sampling

The first thing to note is the **Sampling** setting in the Sampling Tab. The default value is **8 Samples**. Click to display options. Higher values improve the display. Lower values reduce the resolution.

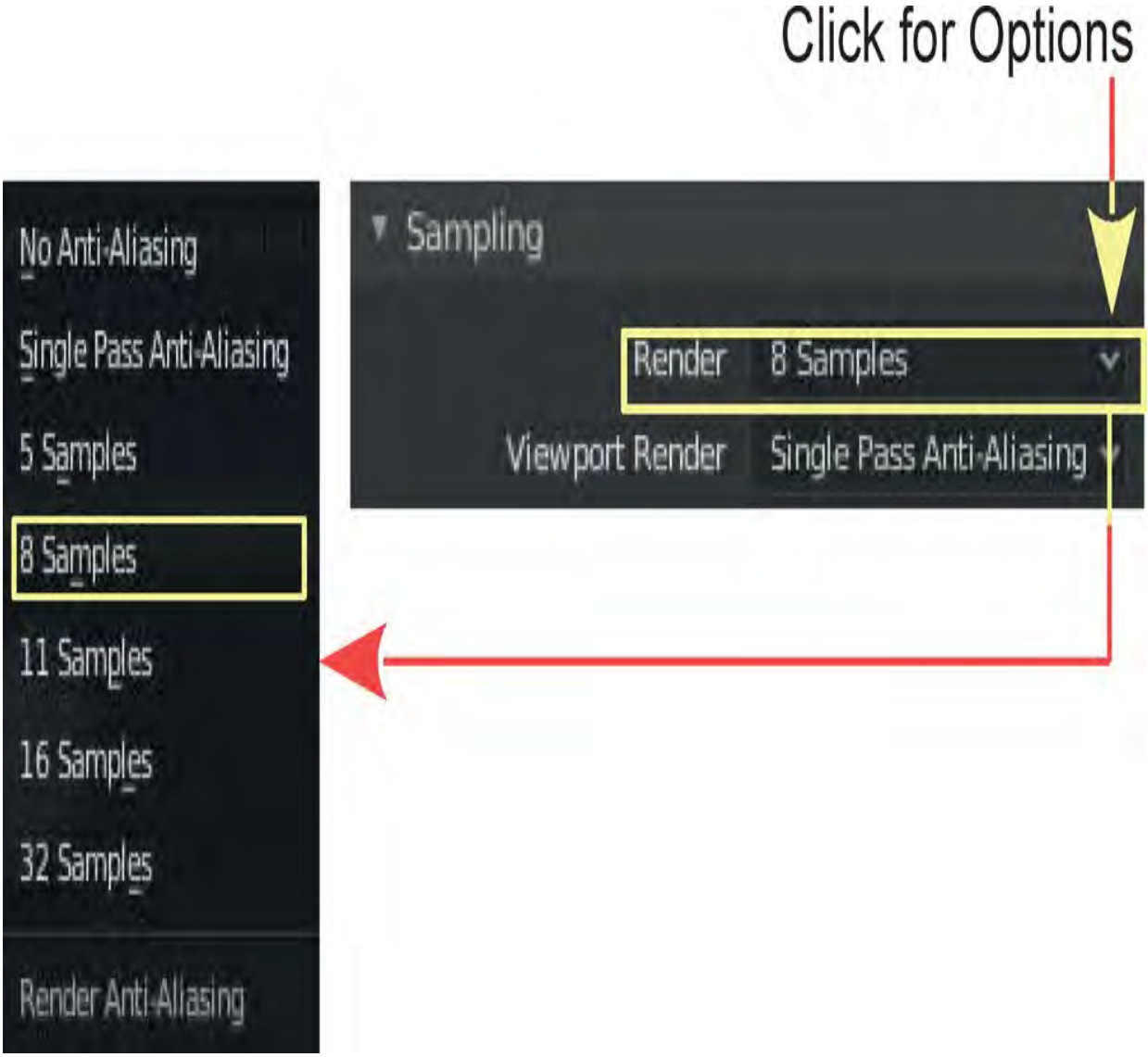


Figure 31.34

Lighting

Scene
Render Engine
Workbench
Sampling
Render
8 Samples
Viewport Render
Single Pass Anti-Aliasing
Lighting
Studio
MatCap
Flat
Rotation:
0°
Color
Material
Object
Vertex
Single
Random
Texture

Figure 31.35

Note: In Workbench Render the default Point Lamp in the Scene or any Lamps placed in the Scene have no effect on lighting. The lighting is controlled by the settings in the Workbench Properties Editor, Render buttons (Figure 31.35).

Studio Lighting: The default Lighting display.

Flat Lighting: Shows all Objects without color in a flat gray.

MatCap Lighting: Provides a variety of options which show the Scene with an overall lighting effect. It should be remembered that the effects are a preview only to allow you to see how the Scene will look after you set the lighting effects in Eevee or Cycles render.

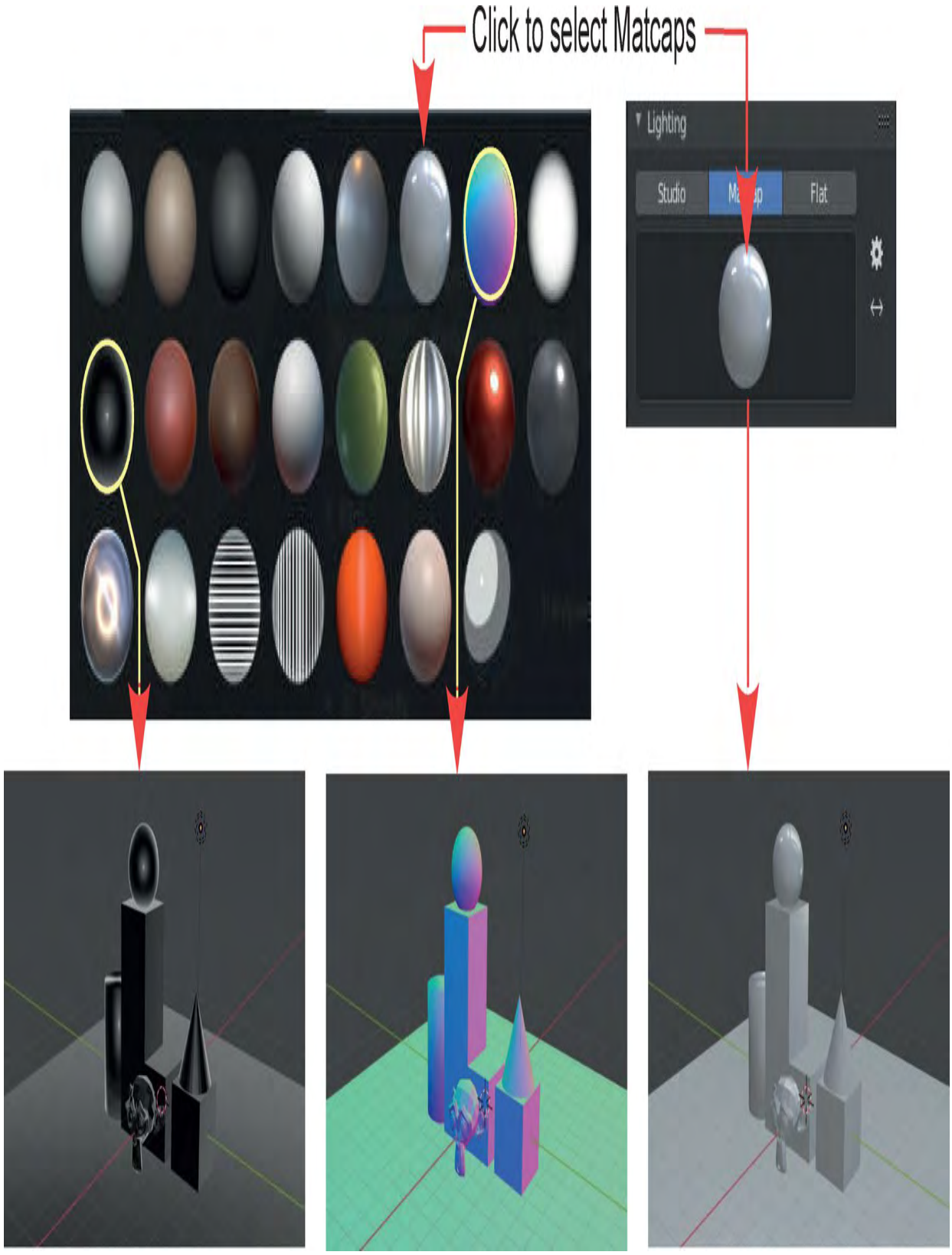


Figure 31.36

Color:

The Color Tab allows you to preview Materials and Textures applied to Objects.

Figure 31.37

The default Color option is **Material** (Figure 31.37). This is perhaps the trickiest option to explain since it relates to the Material applied to an Object and at the beginning of the chapter it was stated that the Material buttons in the Properties Editor were limited.

Material: Assume that you would like to see how Suzanne (Monkey) would look with a different Material, in the Scene,before actually making a change.

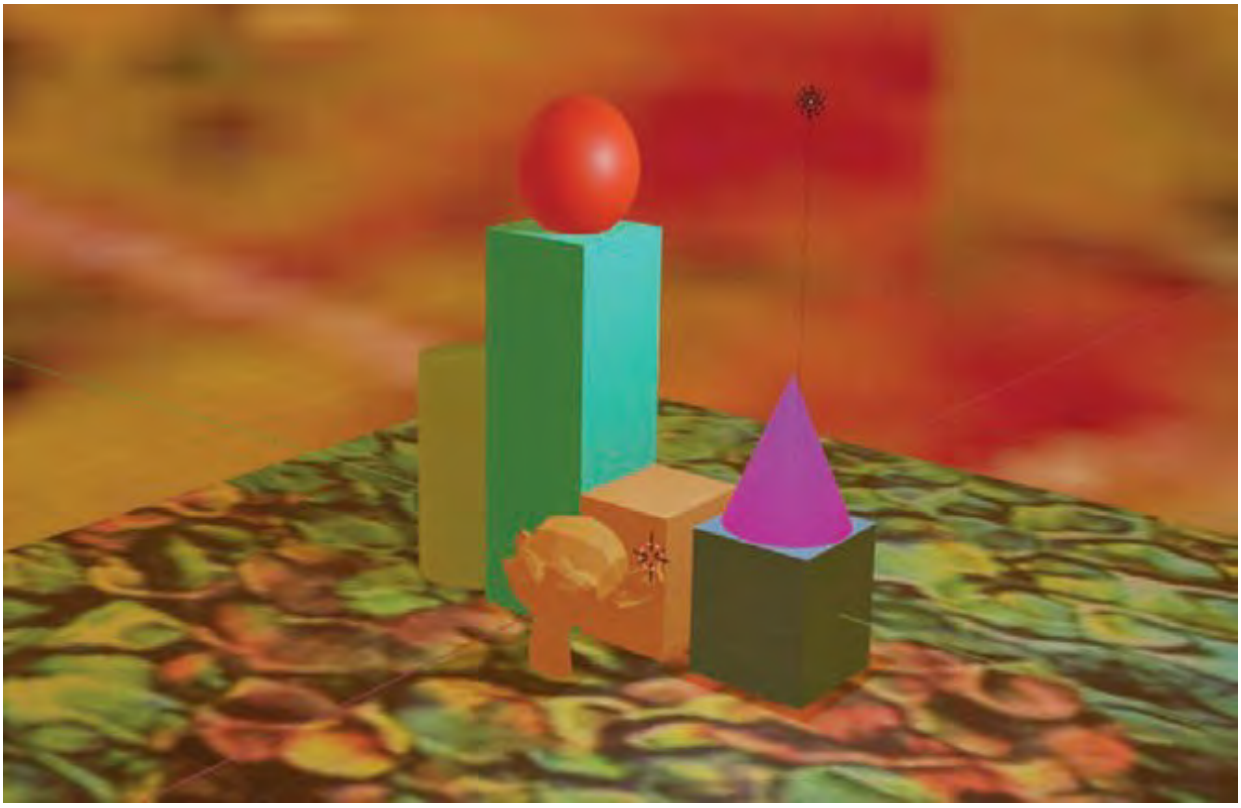


Figure 31.38

The Scene is displayed in Eevee, in Rendered Viewport Shading Mode (Figure 31.38).

Switch to Workbench Render in Rendered Viewport Shading Mode and select Suzanne in the Viewort.

At this point everything in the Scene displays gray. With the limited display in the Material buttons(Figure 31.39), open the Viewport Display Tab and click the Color bar to display the Color Picker Circle. Select a color and adjust the Metallic and Roughness settings (Figure 31.40). When satisfied switch back to Eevee and make the changes.

Object:

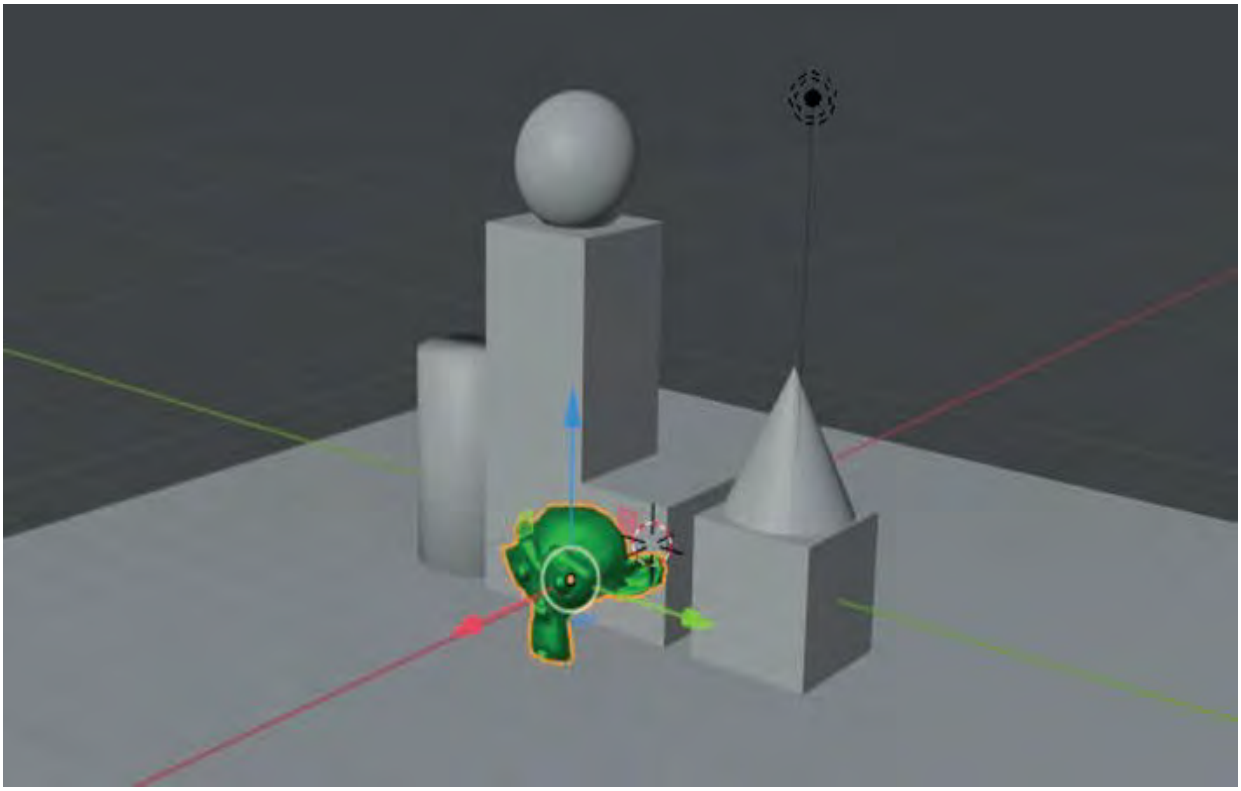


Figure 31.39

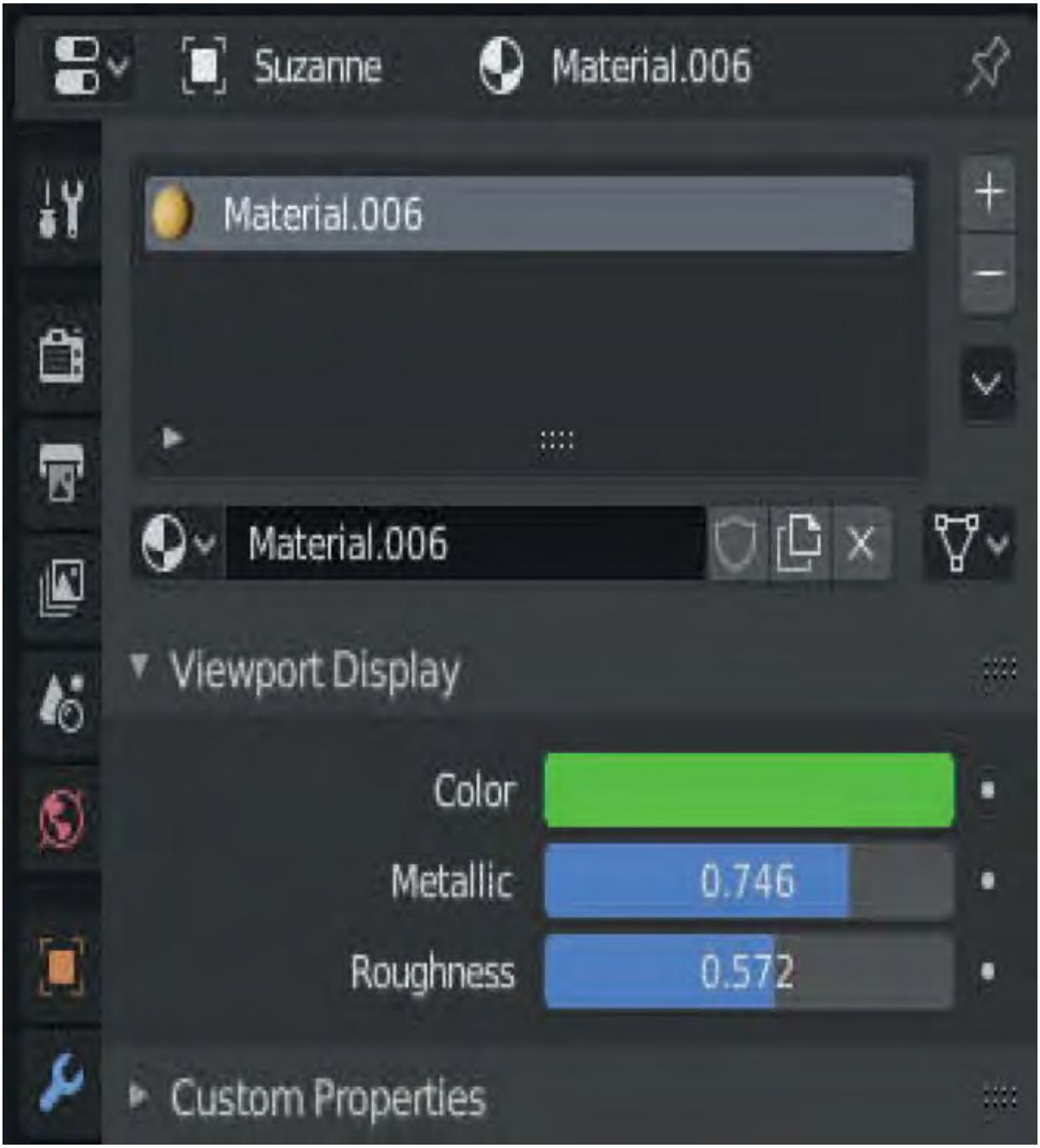


Figure 31.40

The Object Color option is similar to the Material option. Select an Object in the Viewport then in the Properties Editor, Object buttons Viewport Display tab, select a Color for preview.

Single: The **Single** Color option displays all Objects with the same color.

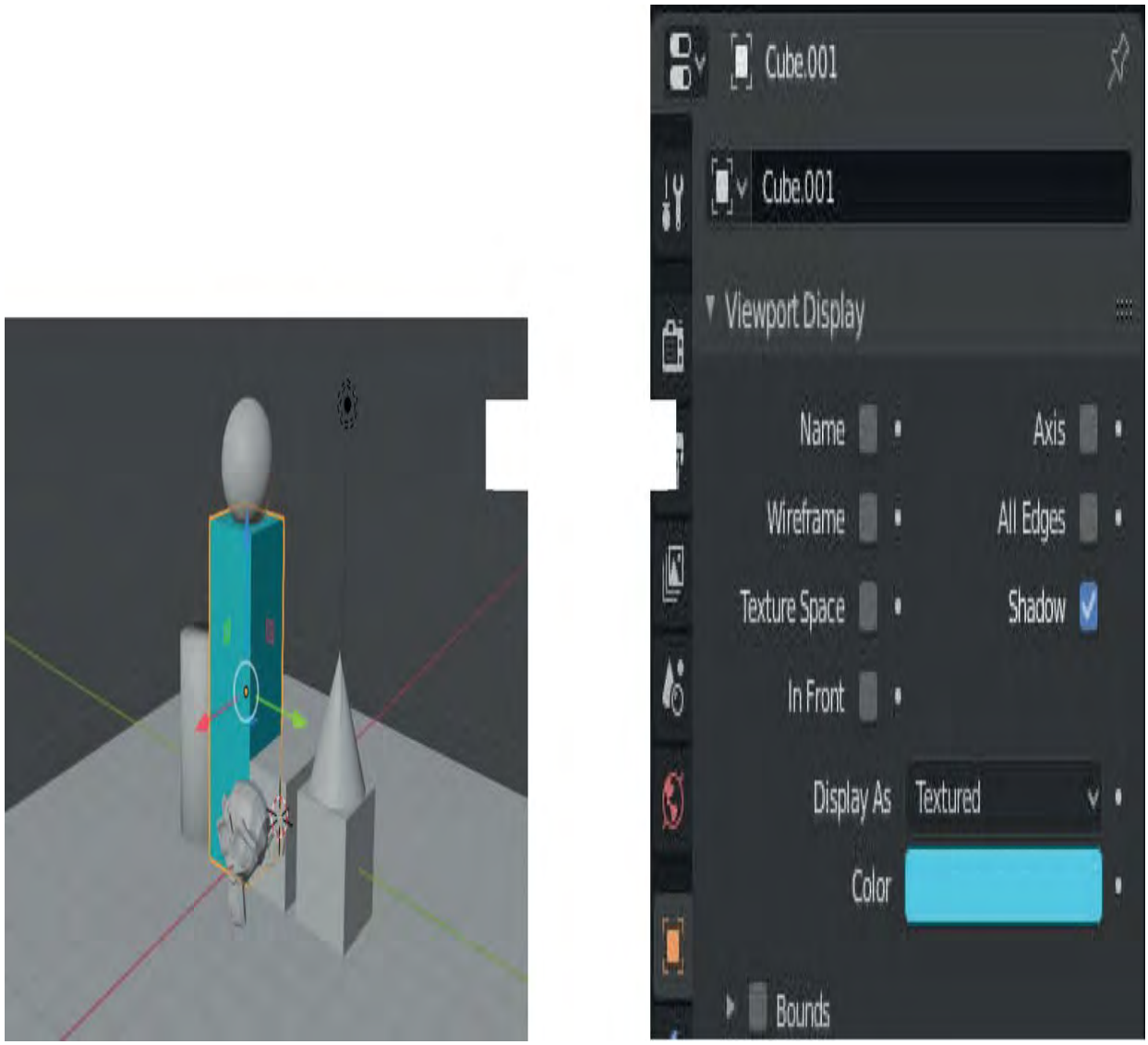

Cube.001
Cube.001
Viewport Display
Name
Axis
Wireframe
All Edges
Texture Space
Shadow
In Front
Display As
Textured
Color
Bounds

Figure 31.41

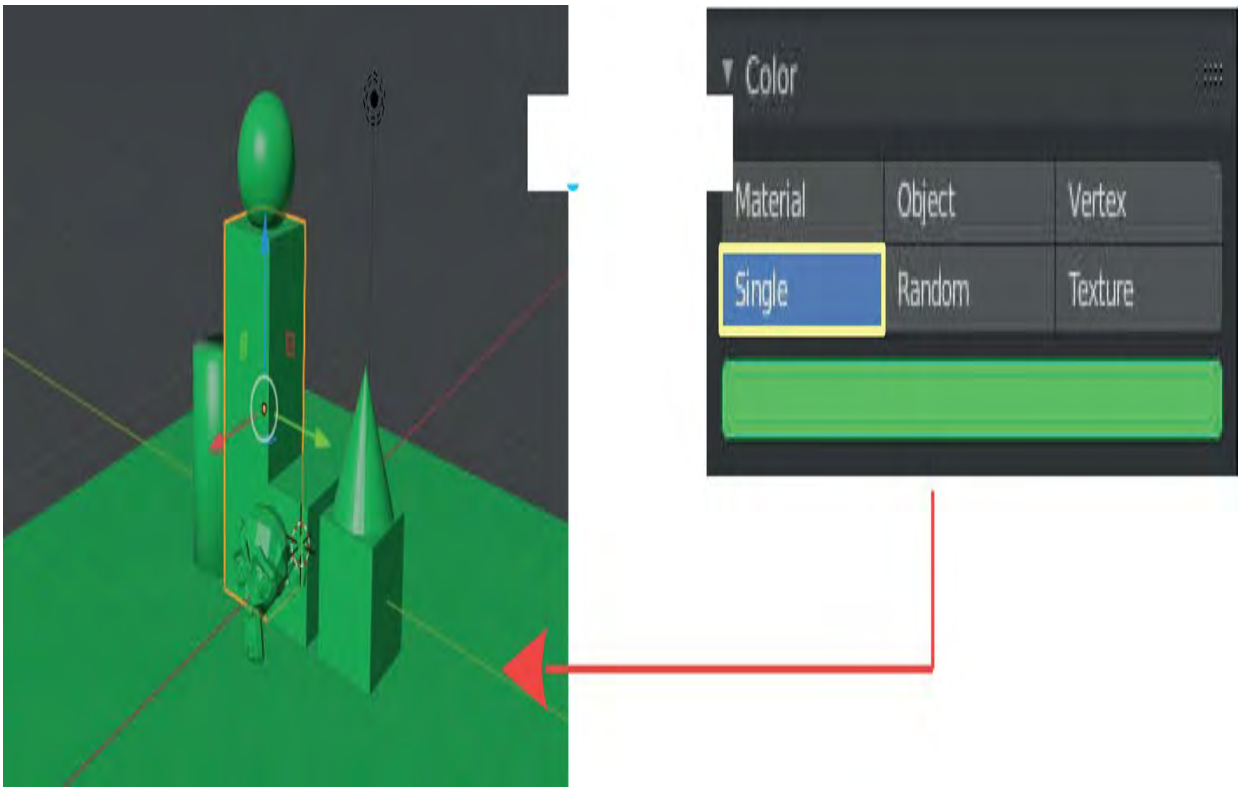


Figure 31.42

Random: Random applies a different color to each Object in the Scene making it easier to distinguish one object from another when the scene becomes really congested.

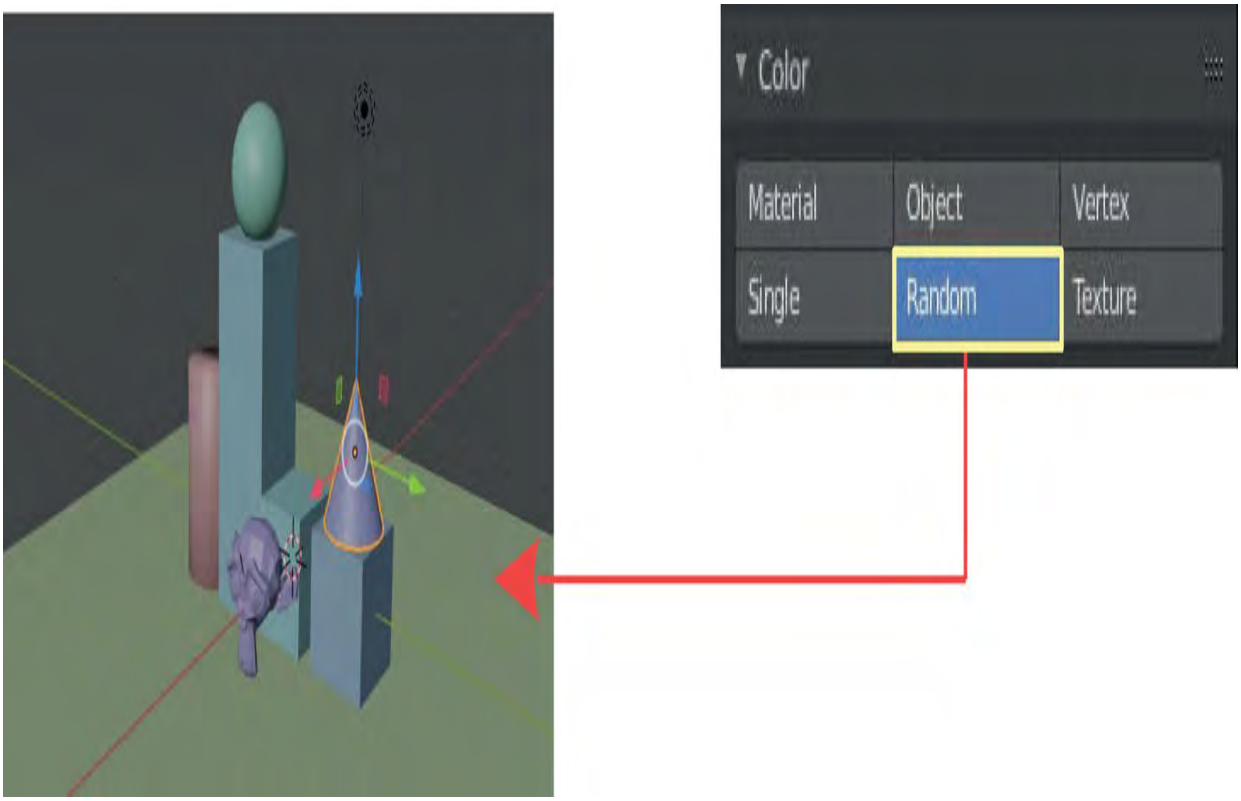


Figure 31.43

Texture: Displays textures which have been applied to Objects in a Scene.

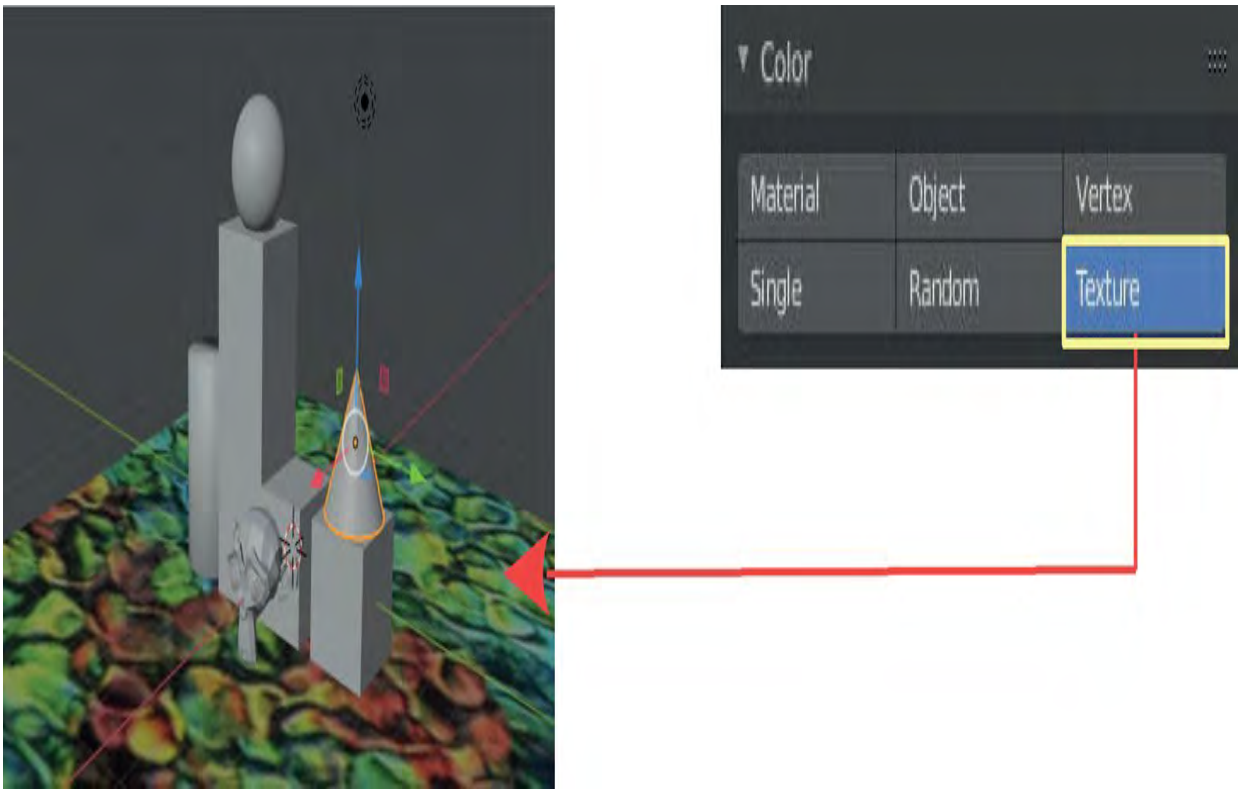


Figure 31.44

Vertex:

Vertex allows you to **Vertex Paint** Objects as a preview before making changes to a Scene. Have the Vertex option selected. Select an Object in the 3D Viewport. Change the 3D Viewport (in Rendered Display Mode) from Object Mode to **Vertex Paint Mode**. RMB click in the Viewport foe the Color Picker. Select a color. The Mouse Cursor is a circle which you click , hold and drag over the Object to paint color.

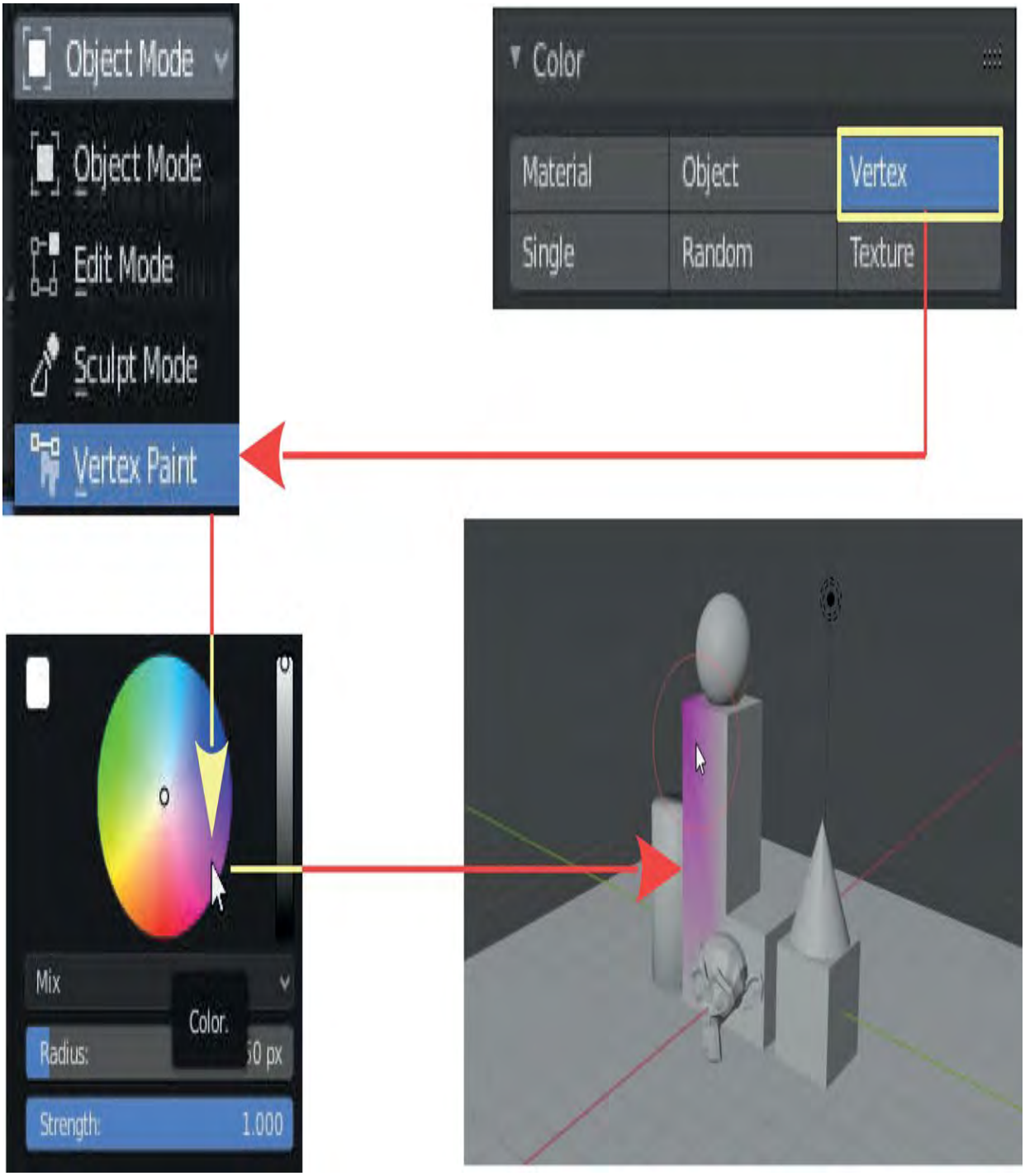
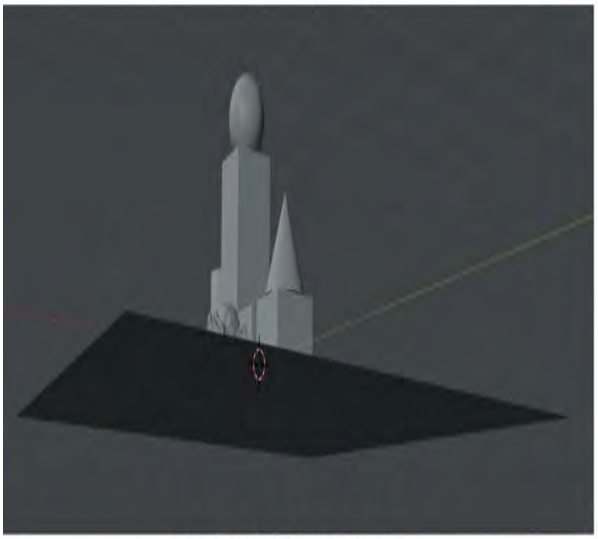


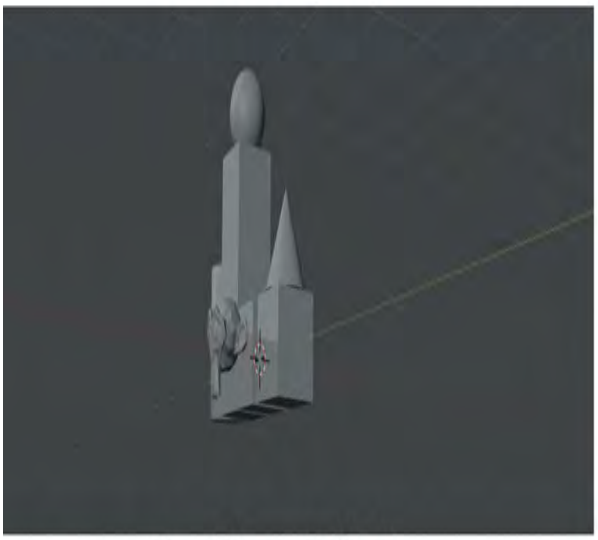
Figure 31.45

The Options Tab

Finally a few of the settings in the **Options Tab**.



Scene Rotated



Backface Culled

Figure 31.46

Backface Culling: Allows you to see through the rear Faceof Objects.

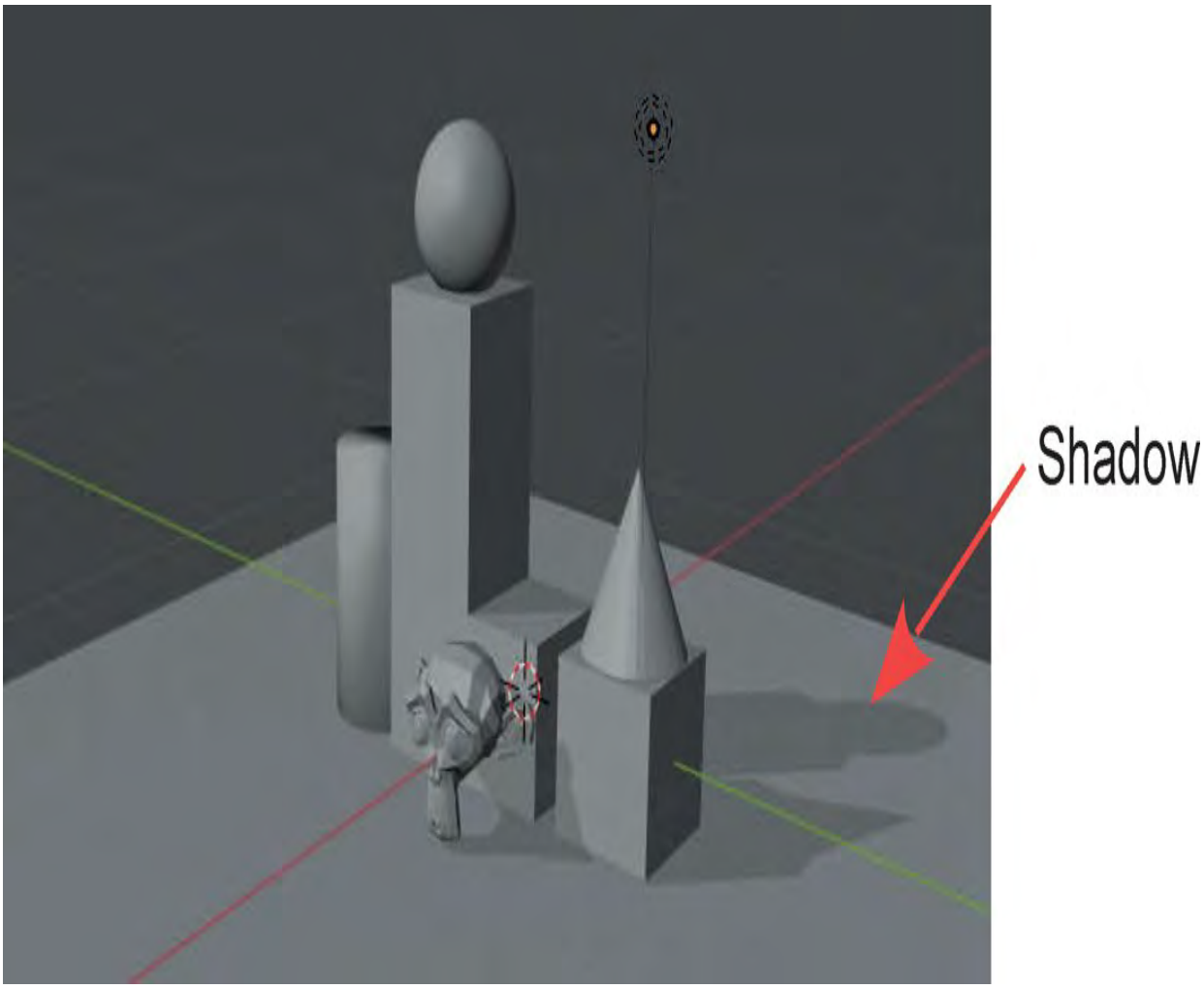


Figure 31.47

X-Ray: Displays Objects in the Scene transparent **IN A RENDERED VIEW** when you press **F12.**

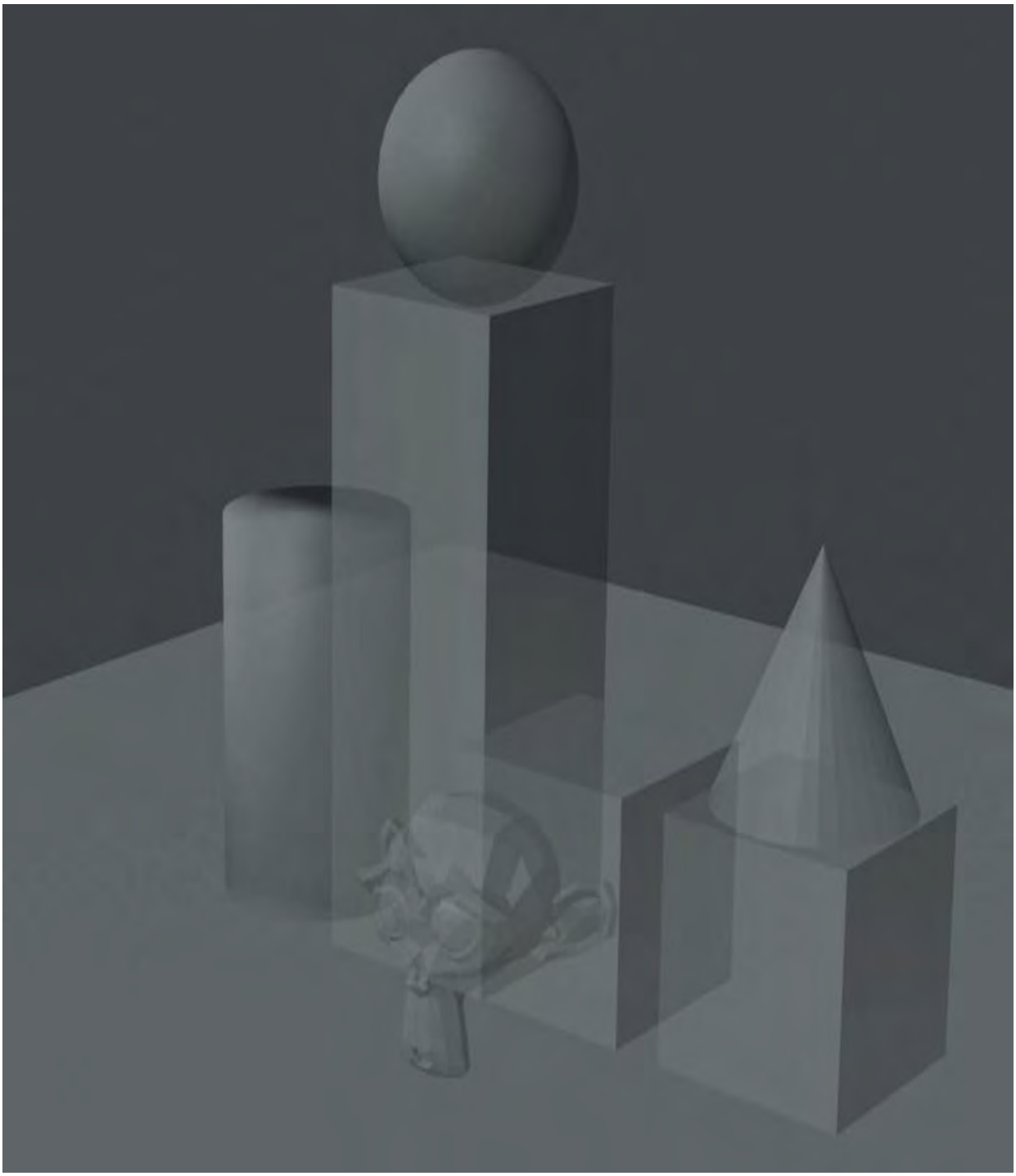


Figure 31.48

Appendix A Basic Blender Commands

This is a partial list of Blender commands. Please visit www.blender.org for more details.

A Key: Press to **select All Object** in the Scene.

LMB Click in the Viewport to deselect All Objects.

B Key: In Object Mode and Edit Mode, press to activate **Box Select** cross hairs; Position the cross hairs with the Mouse, click LMB, hold and drag the Mouse Cursor to create a window around an Object or multiple Objects to be selected. Release the Mouse button to select the Objects.

C Key: In Object Mode and Edit Mode press to activate **Circle Select**; Creates a circle around the Mouse Cursor. Scroll MMB to adjust the diameter. Click LMB, hold and drag the circle over multiple Objects or Vertices to select. Press Esc to cancel leaving the Objects or Vertices selected..

E Key: Extrude; While in Edit mode, selected Vertices, Edges, or Faces, drag Mouse, extrude.

G Key: Grab; press the **G Key** and drag the mouse to freely move a selected Object or selected Vertices. Click LMB to set in position.

I Key: Press to **insert** an animation **Keyframe**.

M Key: Press, with the Mouse Cursor in the 3D Viewport Editor, to display the **Move to Collection** panel.

N Key: Toggles between showing and hiding the Object Properties Panel containing numeric data relating to the selected Object or Vertices.

O Key: In Edit Mode, toggles Proportional Vertex Editing (On - Off).

P Key: In Edit Mode, press to opens the **Separate** menu in order to separate .

R Key: Rotate; Press the R Key and drag the mouse to rotate an Object or selected Vertices.

S Key: Scale; Press the S Key and drag the Mouse to Scale an Object or selected Vertices.

T Key: Toggles the **Tool Panel** display on - off.

U Key: In Edit mode, press to open the **UV Mapping** menu.

W Key: Toggles the Selection Tools – Cursor Select, Box Select, Circle Select, Lasso Select.

X Key: Delete Key. Press to display O.K. Delete menu.

Z Key: Toggles display of Shading option Pie Menu. Click LMB to select an option.

Arrow Keys: Used to advance Frames in an animation: left and right arrows = 1- Frame increments, up and down arrows move to the First and Last Frames.

Ctrl + A Key: Opens the **Apply** menu; this can reset the Object data.

Ctrl + J Key: Joins two selected Objects.

Ctrl + S Key. Opens the **File View** panel to save a Blender file.

Ctrl +Z Key: The global **undo command**; with each press, one step will be undone (up to 32 steps are possible by default). If you are in Edit Mode, it will only undo editing steps on the selected Object.

Esc: Cancels an action or ends an animation.

F2: Opens the Object Name Panel (Edit the Name).

F3: Opens the **Search Panel**.

F4: Opens the **File Context Menu**.

F11: Opens the **Blender Render Editor**.

F12: Renders an image of the Camera view.

LMB: Left Mouse Button . Click to select an Object. Click an empty space to deselect Objects. Click to manipulate the 3D Manipulator Widget, to activate functions, to enter values, etc.

RMB: Right Mouse Button . Click to display the Object Context menu.

MMB: Middle Mouse Button (MSW) . Click to manipulate specified options.

MSW: Mouse Scroll Wheel. Zooms in and out and scrolls to expand/contract selection options.

NumPad: (Number Pad) Controls the view.

7:…. Top

1:…. Front.

3:…. Side.

0:…. Camera.

5:…. Toggles User Perspective – User Orthographic View + and – : Zoom in and out and control the vertex size in proportional vertex editing

Shift Key: Hold down while clicking the LMB to make multiple selections.

Shift + A Key: Displays the **Add** menu to add Objects to the Scene.

Shift + D Key: Duplicates or copies selected Objects or Vertices.

Shift + S Key: Displays the **Snap** pie menu for Selection and Cursor.

Shift + Space Bar: Displays **Tool Panel** options.

Shift + LMB Click: Selects multiple Objects or Vertices.

Shift + RMB Click: Locate the 3D View Editor Cursor. **Note:** No Objects selected.

Space Bar: Toggles Play / Stop Animation.

Tab: Toggles between Edit Mode and Object Mode.

Many shortcuts are listed in the menus accessed in the Editor Headers

3D View Editor Header
Object Mode
View
Select
Add
Object
Toolbar T
Sidebar N
Tool Settings
Adjust Last Operation
Frame Selected Numpad .
Frame All Home
Perspective/Orthographic Numpad 5
Local View
Cameras
Viewpoint
Navigation
Align View
View Regions
Play Animation Spacebar
Viewport Render Image
Viewport Render Animation
Viewport Render Keyframes
Area
All A
None Alt A
Invert Ctrl I
Box Select Tweak Left
Circle Select C
Select All by Type
Select Active Camera
Select Mirror
Select Random
Select More/Less
Select Grouped Shift G
Select Linked Shift L
Select Pattern...
Mesh
Curve
Surface
Metaball
Text
Volume
Grease Pencil
Armature
Lattice
Empty
Image
Light
Light Probe
Camera
Speaker
Force Field
Collection Instance
Transform
Set Origin
Mirror
Clear
Apply Ctrl A
Snap
Duplicate Objects Shift D
Duplicate Linked Alt D
Join Ctrl J
Copy Objects Ctrl C
Paste Objects Ctrl V
Parent
Collection
Relations
Constraints
Track
Link/Transfer Data Ctrl L
Shade Smooth
Shade Flat
Animation
Rigid Body
Quick Effects
Convert
Show/Hide
Clean Up
Delete X
Delete Global Shift X

Appendix B Internet Resources

The following resources are a sample of the information freely available on the internet. New material is continually being released and recognition and appreciation is given to the authors of these publications.

It is practically impossible to cover all aspects of **Blender** in a single publication, therefore, researching information from the internet is integral to becoming proficient in the operation of the program.

In providing the following there is no order of precedence.

Blender 2.93 Reference Manual

https://docs.blender.org/manual/en/latest/

Wikipedia Blender (software)

https://en.wikipedia.org/wiki/Blender (software)

Tutorials — blender.org

https://www.blender.org/support/tutorials/

CGMasters

https://cgmasters.net/free-tutorials/blender-2-8-tutorial-overview-eevee-more/

CGCookie.com Learn to Create 3D Animations, Game Art and Visual Effects with Blender

https://cgcookie.com/learn-blender

Blender Nation

https://www.blendernation.com/category/education/tutorials/videotutorials/

Blenderartists.org

https://blenderartists.org/t/list-of-the-best-training-for-blender-2-8-beta-for-a-complete-newbie/1148435

CGPRESS

https://cgpress.org/archives/cgtutorials/blender-2-8-beginner-tutorials

Blender Guru

https://www.blenderguru.com/

3DPrinting.com

https://3dprinting.com/blender-tutorials/

Sardi' Blender Tutorials

https://www.youtube.com/playlist?list=PLCtL34Gw96S87jqFtVVkZV1FHQyZV4cmn

Blender.Today (Forums)

https://blender.community/c/today

All Blender Tutorials

https://stylizedstation.com/tutorials/blender-tutorials/

The Authors Website

The authors website has supplements to **The Complete Guide to Blender Graphics**.

https://www.tamarindcreativegraphics.com

Supplements:

01	Reflection
02	Background Image
03	Irradience Volume
04	Reflective Plane
05	Background Image
06	Installing Add-ons
07	Drivers
08	Textures in Cycles

Index

2D Animation 559
3D Text 205
3D Viewport Cursor 61
3D Viewport Editor 9, 10, 11, 24

A

Action Constraint 391
Action Editors 441, 447
Add a new Object 9
Add Menu 9
Add Scene 30
Add Theme Preset 44
Adding Bones 400
Adding Collections 201
Add-ons 56
Add-Ons Installing 555
Aircraft Modeling 187
Aligning Bones 410
Amplitude of the Noise 360
Angular Measurement 69
Animating Rotation 361
Animation 17, 331, 366
Animation End Frame 17
Animation Follow Path 369
Animation Play Stop 17

Animation Scrubbing 339
Animation Speed and Length 335
Animation Workspace 382
Aperture 235
Append or Link Command 54
Applying a Material 71
Applying Modifiers 117
Armature Display Types 405
Armature Modifier 146
Armature Rig 398, 401
Armatures 397
Array Modifier 123
Aspect (Aspect Ratio) 323, 324
Attribute Nodes 313
Automatic Keyframing 340
Automatic Weights 418
Awesome Theme Preset 43

B

Background Color Lighting 241
Background Displays 222
Background Image 235
Baking 516, 520
Base Color 14, 18
Basis Key 443
Bendy Bone Handles - Add-On 555
Bevel 101
Bevel Modifier 122, 115

Bezier and Nurbs Circles 173
Bezier Circle 171
Bezier Curve 170
Bezier Interpolation 334
Bfont 208
Blend Swap 426
Blender Logo 4
Blender Node System 258
Blending Material Color 302
Boid Brain 475
Boids Particle 475
Bone Allignment 410
Bone Armature 398
Bone Chain Length 422
Bone Constraint Properties 421
Bone Naming 408, 434
Boolean Modifiers 125
Border Select 62
Bounding Volume 545
Brightness Slider 14, 72
Browse Material to be Linked Button 251
Browse Texture to be Linked 297
Build Modifier 126

C

Cache File Path 517
Cage Mesh 156
Camera Switching 237

Camera Tracking 239
Camera View 28, 227
Cameras 227, 232, 234
Cancel an Editor 24
Cast Modifier 146, 380
Center of Geometry 87
Center Points 87
Character Rigging 425
Chevrons 179
Chibi Model 425, 426
Child Parent Relationship 96, 397, 401, 404
Circle of Influence 93, 94
Circle Select 63
Close the Program 4
Closed Loops 173
Cloth Physics 509, 510
Codecs 330
Collada File 55
Collection Properties 203
Collections 75, 199
Collections in the Outliner 31
Collision and Outflow Objects 524
Color Gradient 272
Color Palette 279
Color Picker 14, 255
Color Ramp Node 270
Color Schemes 14
Coloring 13

Coloring Objects 71
Coloring using Texture 303
Combine RGB Node 258
Compositing Nodes 317
Compositing Workspace 318
Constant Extrapolation 358
Constraint Stack 385
Constraints 383, 384
Converting Text to a Mesh 212
Copy and Paste 76
Corrective Smooth Modifier 147
CUDA™ 590
Cursor 9, 61
Cursor Location 64
Cursor Pie Menu 64
Curve Modifier 148
Curve Properties Panel 360
Curve Types 357
Curves Circles and Paths 168
Cutting Video Strips 570
Cycles in Practise 594
Cycles Render 587
Cyclic Extrapolation 358

D

Data 57
Decimate Modifier 128
Deform Modifiers 145

Deforming a Mesh 412
Deleting 19
Deleting Collections 202
Deleting Objects 65
Depth of Field 235
Displace Modifier 149, 303
Displacement Sound Animation 376, 380
Display Options 52
Domain Bounds 545
Dope Sheet Action Editor 449
Dope Sheet Editor 341, 342
Dope Sheet Timeline 353, 356
Drivers 575
Drivers Editor 577
Duplicating Objects 65
Dupliverts 185
Dynamic Paint 547

E

Edge and Loop Selection 102
Edge Loop 102
Edge Slide 111
Edge Split Modifier 129
Edit Mode 11, 61
Edit Mode Tool Panel 98
Editing Objects 79
Editing Techniques 183
Editing Tools 97

Editor Modes 11
Editor Types 22
Editors 1, 6, 21
Eevee 321
Eevee Render Engine 59
Effector - Collision 534
End Frame - Animation 17
Envelope Display 411
Envelope Weights 417
Environment Texture 243
Esc to Quit 28
Euler Rotation 350
Evaluation Time 371
External Data 57
External Font 213
Extrapolation 358
Extrude Region Tool 99, 100
Extrude Tool 411
Extrude/Bevel 394

F

F key to Face 85
F-Curve Rotation 364
File - Opening 51
File Browser 47, 48, 52
File Concept 54
File Display Options 52
File Extensions 46

File Saving 47
File Type Import 56
Files and Folders 46
Filling a Cup 527
Fire - Quick Method 15
Fire Simulation 17
Flock of Birds 475
Fluid Flow Behaviour 534
Fluid Simulation 513
Fluid Viscosity 529
Fly Navigation 29
Follow Path 394
Fonts 208
Force Field 473, 509
Frame Rate 335
Frame Start-End-Step (Frame Range) 325
Functions 6

G

General Workspace 21
Generate Modifiers 121
Geometry Node Editor 308
Gizmos and Overlays 25
Gradient Colors 38
Gradient High/Off 38
Graph Editor 343, 348, 349
Graph Editor Cursor 351, 352
Graphical User Interface (GUI) 1, 2, 6, 21

Grease Pencil 2D Animation 559
GUI 2

H

Hair Color 484
Hair Particles 479
HDRI Maps / Images 242, 243
Header Controls 34
Headers 6
Help 5
Hiding and Restricting Object Display 202
Hook Modifier 150
Hook-Empty 152, 153
Humanoid Figure 195

I

Image as a Texture 285
Image Texture 281
Images as a Background 242, 245
Images for Tracing 244
Import File Types 56
Importing Objects 55
Insert Keyframe 18
Inset Faces 94, 101
Installing Add-Ons 555
Instance 311
Interface 1, 6
Interface Header 7

Interpolation 334
Inverse Kinematics 420, 437

J

Joining and Separating 88, 96
Joining Objects 76

K

Key Slider 443
Keyed Particles 502
Keyed Physics 502
Keyframe 18, 337
Keyframes Time and Interpolation 334
Keyframing 373
Keying Sets 367
Kinematics 421
Knife Tool 104

L

Landscape 184
Laplacian Deform Modifier 152
Last Operator Panel 66, 84, 93, 95, 99
Lathing 178
Lattice Modifier 153
Layout Workspace 30
Light Type 228
Lighting 227, 231

Lighting Affects 256
Lighting Arrangement 239
Lighting Color Display 220
Lighting Tones 219
Limiting Bake Time 519
Limits of Movement 444
Linear Extrapolation 358, 363
Linear Measurement 69
Link 54
Lofting 178
Loop Cut 102
Low Poly Model 425

M

Make Human 55, 406
Make Human Program 57
Making a Movie 330, 561
Manipulation Units 67
Manipulation Widget 29, 67, 82
Mantaflow 513
Mapping 287
Mask Modifier 130
MatCap 220
Matcap Lighting 220, 225
Material – Applying 71
Material 14
Material Assignment 252
Material Cache 251, 252

Material Capture 220
Material Nodes 15
Material Preview Shading 225, 229
Material Properties 13, 72, 251
Material Property Drivers 585
Material Slot 251, 253
Materials – Textures – Nodes 247
Materials 248
Materials using Nodes 257
Measurement 69
Measuring Ruler / Protractor 69
Memory Computer Power and Baking 516
Mesh Deform Modifier 156
Mesh Deforming 412
Mesh Object 80
Meta Shapes 77
Metashapes 74
Metric Units 67
Mid Plane Grid 7
Mirror Modifier 131
Mirror Sculpt 193
Mist Effect 236
Mixing Material 266
Mode Switching 61
Model – Low Poly 425
Modeling 12
Modeling a Bowl 531
Modeling a Cup 535

Modeling from a Curve 172
Modeling Workflow 60
Modifier Stack 118
Modifiers – Generate 121, 122
Modifiers 113
Modifiers Properties 114
Monkey 9
Mouse Cursor 2, 4, 61
Movement in the 3D View Editor 332
Moving in 3D Space 27
Multiple Material Datablocks 252
Multiple Particle Systems 500
Multiple Scenes 30
Multiresolution Modifier 133

N

Naming Bones 408
Naming Objects 74
Navigate and Save 45
Navigation 29
Node Effects 259
Node Noodle 262
Node Pipeline (Tree) 310
Node Socket 262
Node System 14, 258, 305
Node Wrangler 295
Nodes 258
Nodes Active – Nodes Cancelled 73

Nodes in the Shader Editor 261
Noise Amplitude 360
Noodle 262
Noodle Curving 263
Normals – Particle Emission 465
Normals 169
Nurbs Circle 171, 175
Nurbs Curves 174, 178
Nurbs Path 171

O

Object Context Menu 87
Object Editing 79
Object Light Source 592
Object Lighting 600
Object Mode – Edit Mode 11
Object Mode and Edit Mode 61
Object Mode Manipulation 65
Object Naming 74
Object Offset 124
Object Primitives 63
Object Properties Panel 35
Object Select – Deselect 62
Object Types 74
Objects 8, 59
Offset Object 124
Opening Files 51
Options in the Outliner Editor 203

Orthographic View 27, 86
Outliner and Collections 199
Outliner Editor 9
Overlays 26
Overlays and Gizmos 25

P

Packing Data 57
Paint Layers 276
Painting with a Texture 301
Parenting 96
Parenting Armatures 413
Particle Cache 457
Particle Color 471
Particle Color Velocity and Acceleration 478
Particle Display 461
Particle Emission 457, 464
Particle Instance Modifier 468
Particle Interaction 472
Particle Settings 455
Particle System Modifier 468
Particle Systems 453, 455
Particle Velocity 459
Particles Array 469, 489
Path 168
Path Animation 370
Path Length Graph 376
PBR Textures 281, 290, 295

Perspective View 27, 86
Physics – Quick Methods 525
Physics and Simulation 505
Pie Menu 64
Playing the Animation 339
Poly Build Tool 105
Pose Mode 402
Posing the Character Model 437
Power of Nodes 598
Precise Manipulation 70
Preferences Editor – Add-ons 56
Preferences Editor 37, 38
Preferences Editor Presets 42
Presets 42
Primitive Object 59
Primitives 8
Primitives for Selection 64
Principled BSDF Node 267
Principled Volume Node 316
Procedural Textures 281, 282, 284
Properties Editor 13, 35, 36
Proportional Vertex Editing 93, 184
Python 259
Python Script 580

Q

Quick Method 15
Quick Methods – Physics 525

Quick Smoke 16
Quick Smoke Method 539
Quit 5

R

Radians 69
Ramp Controls 270
Random Color Display 221
Randomize 110
Randomize Object Properties 580
RandomPy.blend File 583
Real World Physics 506
Reference Image 244
Relationship Constraint 391
Relative Offset 124
Remesh Modifier 133
Render 28
Render Engines 587
Render Farm 330
Render Format 334
Render Output Properties 323
Render Properties 322
Rendered Viewport Shading 215, 224
Rendering 321, 322
Rendering a JPEG Image 326
Rendering a Movie File 327
Rendering a Video Clip 327
Rendering the Movie File 571

Resizing Editors 22
Resolution 323
Restart 5
RGB HSV Hex 14
RGB Values 14
Rig Construction 407
Rigging a Character 425
Rigid Body Physics 507, 508
Rip Region 112
Rotating in the 3D Space 7
Rotation – Euler Rotation. 350
Rotation 10
Rotation Animation 361
Rotation Using F-Curves 364

S

Saving a File 47
Saving Screen Space 273
Saving Work 45, 53
Scene 30
Scene Collection 31, 200
Scene Lighting 227, 228, 241
Scene Manipulation 26
Scene Properties 67
Scene Selection 31
Screw – Tool – Modifier 109
Screw Modifier 134
Scrubbing the Animation 339

Sculpt Mode 190, 192
Sculpting 190
Sculpting Screen 2
Select Box Button 62
Selecting – Deselecting Objects 62
Selection Circle 63
Selection Mode 12
Separate by Loose Parts 90
Separate Menu Options 90
Separating in Edit Mode 89
Set Parent To Object 96
Shade Smooth 76
Shader Editor 257
Shader Node Menu 264
Shading Modes 81
Shading Workspace265
Shadow 219
Shape Key 441, 449
Shear 111
Shrink Fatten 111
Shrinkwrap Constraint 392
Shrinkwrap Modifier 157
Simple Deform Modifier 158
Simplified Viewport Shading 226
Single Bone Armature 398
Skin Modifier 136
Sky Lighting 599
Slide 111

Smoke 16
Smoke and Fire Simulation 537, 538
Smoke from Scratch 540
Smooth and Flat Shading 76
Smooth and Randomize 110
Smooth Corrective Modifier 160
Smooth Laplacian Modifier 160
Smooth Modifier 159
Snap Tool Pie Menu 64
Sockets 262
Soft Body Physic 512
Solid Viewport Shading 218
Solidify Modifier 137
Sound Effect 380
Sound File 563
Speed of Rotation 263
Spin 105
Spin Duplication 108
Spin Profile 106
Splash Screen 2, 3, 7
Spline IK Constraint 423
Split Area 23
Splitting Editors 23
Spotlight Size Animation 367
Spreadsheet Editor 310
Start Over 5
Studio Lighting 218
Subdivision 83

Subdivision Surface Modifier 138
Surface Deform Modifier 161
Suzanne 8
Switching Modes 61

T

Tangent Phase - Particles 460
Text 205
Text Fonts 208
Text on a Curve 211, 212
Text to Mesh 212
Texture Cache 303
Texture Color Display 222
Texture Displacement 303
Texture Image 285
Texture Mapping 287
Texture Node Editor 306
Texture Paint Mode 306
Texture Painting 299
Texture Properties 297
Textures 281
Theme 21, 41
Theme Space 38
Thumbnails 52
Timeline Editor 336
Timeline Graduations / Positions 568
Toggle Button 26
Toggle X -Ray function 81

Tool Panel - Widgets 67
Tool Panel 35
Tracing Images 244
Track to Constraint 384, 389
Transform Constraints 386, 387
Translate, Rotate and Scale 65
Transparency 314
Triangulation and Weld Modifier 140
Types of Curves 357

U

Uncluttering the Scene 25
Undo Operation 19
Unit System 67
Unwrapping Objects 289
Use Nodes 14
User Perspective and Orthographic View 27
User Perspective View 86
UV Editor 288

V

Vector Font for Text Objects 208
Vertex Color Display 222, 227
Vertex Editing 93
Vertex Groups 91, 436
Vertex Paint 274
Vertex Paint Layer 277
Vertex Slide 111

Vertices by Subdivision 83
Vertices Edges Faces 11, 79
Video Codecs 330
Video Editing Workspace 563
Video Files 562
Video Format 565
Video Playback 329
Video Sequence Editor 561, 566
Video Strip - Cutting 570
View - Navigation 29
View Layer 203
Viewport Features 24
Viewport Gizmos 25
Viewport Shading 215
Viewport Shading Modes 15, 81
Viscosity 529
Volumetric Lighting 246, 315

W

Walk Cycle 439
Walk Navigation 29
Warp Modifier 162
Wave Modifier 164
Wavefront OBJ 57
Weigh Paint 91, 418
Weld Modifier 140
Whale’s Tooth 143
Widget 10, 29, 34, 67

Wind Force 473
Windows 10 File Explorer 46, 48, 49
Windows Fonts 208
Wireframe Colors 217
Wireframe Display 82
Wireframe Modifier 143
Wireframe Viewport Shading 216
Workbench Render 587, 601
Workspace 21, 30, 33. 39
World displays 223
World Properties 223
World Settings 226

X

X Key Delete 19
X Key or Delete Key 85